

---

Master Thesis

# **Vollautomatisches Mosaikieren von Satellitenbildern**

Philipp Serr

---



30. September 2013

b

---

**Master Thesis**

# **Vollautomatisches Mosaikieren von Satellitenbildern**

Philipp Serr

Institut für Informatik  
Ludwig-Maximilians-Universität München

Sommersemester 2013

Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)  
Earth Observation Center (Oberpfaffenhofen)

Technische Universität München (TUM)  
Computer Graphics and Visualization Group, Prof. Dr. Rüdiger Westermann

Betreuer: Prof. Dr. Rüdiger Westermann (TUM), Dr. Tobias Storch (DLR)



---

## Kurzfassung

---

Bei der Analyse weiträumiger Gebiete soll häufig eine große Anzahl an Satellitenbildern, die mit optischen Erdfernerkundungssatelliten aufgenommen wurden, zu einer Mosaikdarstellung zusammengesetzt werden. Dazu muss jeder Bildpunkt des Mosaiks aus den Pixeln der Einzelaufnahmen abgeleitet werden. Sind mehrere Aufnahmen vorhanden, die den gleichen Bildpunkt enthalten, muss entweder eine Einzelaufnahme ausgewählt werden, die das Pixel exklusiv beiträgt oder der Intensitätswert des zugehörigen Mosaikpixels aus den Pixeln mehrerer Aufnahmen errechnet werden. In dieser Arbeit werden Methoden betrachtet, die jedes Pixel des Mosaiks genau einer Aufnahme zuordnen und somit die ursprünglichen Intensitäten (Pixelwerte) erhalten. Schwierigkeiten ergeben sich aus geometrischen und radiometrischen Unterschieden der beteiligten Aufnahmen. In jedem Fall gilt es große Datenmengen zu verarbeiten. Änderungen des zugrundeliegenden Datensatzes durch das Hinzufügen oder Entfernen von Aufnahmen sollten daher keine vollständige Neuberechnung des Mosaiks erfordern. Diese Arbeit nähert sich dem Problem von verschiedenen Seiten: Zum einen wird die Problemstellung als kombinatorisches Optimierungsproblem aufgefasst, das sich mit Graphenalgorithmen in polynomieller Zeit lösen lässt. Auf der anderen Seite vertieft die Arbeit einen heuristischen Ansatz, der darauf abzielt, Bereiche des Mosaiks, die unterschiedlichen Einzelaufnahmen zugeordnet werden, entlang von Geländestrukturen abzugrenzen. Die vorgeschlagene Methode skaliert linear. Beide Verfahren wurden in Software umgesetzt und anhand realer Aufnahmen unterschiedlicher Missionen getestet.



---

# Inhaltsverzeichnis

---

<b>Kurzfassung</b>	<b>3</b>
<b>Inhaltsverzeichnis</b>	<b>5</b>
<b>1 Überblick</b>	<b>7</b>
1.1 Einleitung . . . . .	7
1.2 Ausgangssituation, Anforderungen und Ziele . . . . .	7
1.3 Verwandte Arbeiten . . . . .	12
1.4 Aufbau der Arbeit . . . . .	15
<b>2 Einfache Verfahren</b>	<b>17</b>
2.1 Einfaches Übereinanderlegen . . . . .	17
2.2 Voronoisegmentierung . . . . .	18
<b>3 Pixel Labeling - Das optimale Mosaik?</b>	<b>21</b>
3.1 Das Mosaik als Graph . . . . .	22
3.2 Minimale Schnitte - Binäres Labeling . . . . .	23
3.3 Max-Flow Min-Cut . . . . .	25
3.4 $\alpha$ -Expansion Algorithmus für mehr als zwei Label . . . . .	26
3.5 Kostenfunktion . . . . .	27
3.6 Iteratives Mosaikieren . . . . .	29
3.7 Verringern der Komplexität . . . . .	31
<b>4 Morphological Image Compositing</b>	<b>35</b>
4.1 Vorüberlegungen . . . . .	35
4.2 Algorithmus . . . . .	37
4.3 Vor- und Nachteile . . . . .	43
<b>5 Implementiertes Verfahren: MSeam</b>	<b>47</b>
5.1 Von <i>Morphological Image Compositing</i> zu <i>MSeam</i> . . . . .	48
5.2 Iteratives Mosaikieren . . . . .	49
5.3 Das Maskenbild . . . . .	51
5.4 Vollautomatisches Mosaikieren . . . . .	62
5.5 Maskieren unerwünschter Bildausschnitte . . . . .	63
5.6 Renderschritt und Filtern von Artefakten . . . . .	64
5.7 Vor- und Nachteile: Vergleich mit <i>Morphological Image Compositing</i> . . . .	65

<b>6</b>	<b>Ergebnisse</b>	<b>69</b>
6.1	Hochaufgelöste panchromatische Aufnahmen . . . . .	69
6.2	RGB-Mosaik . . . . .	71
6.3	Image-2006 - Mosaikieren großer Datensätze . . . . .	71
6.4	Laufzeit . . . . .	77
6.5	Weitere Verfahren . . . . .	77
<b>7</b>	<b>Fazit</b>	<b>81</b>
<b>A</b>	<b>Implementierung</b>	<b>83</b>
A.1	FAMOS - Ein kurzer Überblick . . . . .	83
A.2	Polymask - Pixelmasken effizient als Polygon darstellen . . . . .	84
A.3	Wasserscheidentransformation . . . . .	85
<b>B</b>	<b>Hinweise zur Benutzung</b>	<b>87</b>
B.1	XML-Projektbeschreibung . . . . .	87
B.2	Implementierte Filter und Algorithmen . . . . .	89
	<b>Abbildungsverzeichnis</b>	<b>93</b>
	<b>Software &amp; Bibliotheken</b>	<b>95</b>
	<b>Bibliografie</b>	<b>99</b>
	<b>Eigenständige Anfertigung der Arbeit</b>	<b>103</b>



# Überblick

---

## 1.1 Einleitung

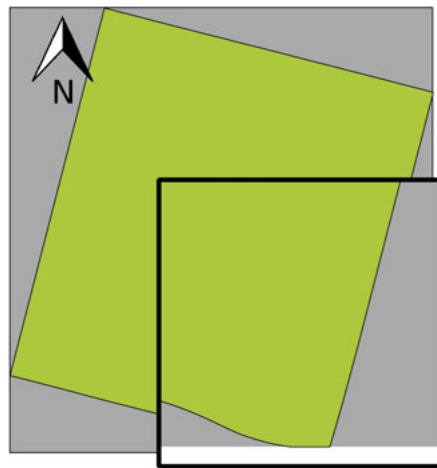
Luft- und Satellitenbilder stellen eine wichtige Datenbasis der Erdfernerkundung dar. Insbesondere werden daraus Orthofotos abgeleitet. Das sind fotografische Bilder der Erdoberfläche mit den geometrischen Eigenschaften einer topografischen Karte. Der breiten Öffentlichkeit sind solche Aufnahmen als Kartenhintergrund in beliebten Online-Kartendiensten bekannt. Hierbei zeigt sich die Notwendigkeit, ein Mosaik aus vielen Einzelaufnahmen zusammenzusetzen besonders deutlich. Auch lässt sich bei Satellitenaufnahmen ein gewisser Wolkenanteil häufig nur schwer vermeiden. Indem mehrere Einzelaufnahmen zusammengefügt werden kann dennoch eine annähernd wolkenfreie Darstellung erreicht werden.

Unter Mosaikieren versteht man den Gesamtprozess der Herstellung eines Orthofotomosaiks aus teilweise überlappenden und zunächst unbearbeiteten Einzelaufnahmen [1], [2]. Dabei wird zunächst eine Serie von Orthofotos (siehe 1.2.1) in der gewünschten Auflösung und Kartenprojektion generiert. Das Mosaik wird dann in einem abschließenden Compositing-Schritt zusammengesetzt. Dabei gilt es für jedes Pixel des Mosaiks einen Wert aus den verfügbaren Einzelaufnahmen abzuleiten. In dieser Arbeit werden Verfahren beschrieben, die jedes Pixel des Mosaiks eindeutig, also jeweils genau einem Ausgangsbild zuordnen. Benachbarte Pixel, die unterschiedlichen Aufnahmen zugeordnet werden, bedingen im Allgemeinen störende Artefakte, sogenannte Nahtlinien. Angestrebt wird daher eine möglichst unscheinbare Platzierung der Nahtlinien.

## 1.2 Ausgangssituation, Anforderungen und Ziele

### 1.2.1 Ausgangssituation

Die Arbeit geht von bestimmten Eigenschaften der zugrundeliegenden Bilddaten aus. Dabei handelt es sich um sogenannte Orthofotos, also aufbereitete Luft- oder Satellitenbilder mit den geometrischen Eigenschaften einer topografischen Karte. Im Gegensatz zur Karte zeigt das Orthofoto jedoch keine vereinfachte Geländedarstellung sondern bildet die tatsächliche Oberflächentextur ab. Man unterscheidet zwischen *panchromatischen* Aufnahmen mit nur einem breitbandigen Kanal, *Multispektralaufnahmen*, die mehrere Kanäle umfassen sowie *Hyperspektralaufnahmen* mit sehr vielen schmalbandigen Frequenz-



**Abbildung 1.1:** Der Definitionsbereich einer Szene (grün) ist im allgemeinen nicht rechteckig. Für die meisten Projektionen ergibt sich ein trapezförmiger Definitionsbereich, wobei die Kanten allerdings nicht gerade verlaufen müssen (Detailansicht). Aus technischen Gründen wird das minimal umgebende Rechteck gespeichert (grau).

kanälen. In jedem Fall muss die Aufnahme nicht auf den sichtbaren Frequenzbereich beschränkt sein und kann sogar ausschließlich Mikrowellenkanäle (Radar) umfassen [1]. Die Darstellung erfolgt gegebenenfalls als Falschfarbenbild. Diese Arbeit geht von panchromatischen oder multispektralen, optischen Aufnahmen aus. Satellitenbilder werden in verschiedene Produktklassen eingeordnet. Die Klassifizierung ist nicht standardisiert und unterscheidet sich von Anbieter zu Anbieter. In Tabelle 1.1 sind die von der ESA für die Sentinel-2-Mission verwendeten Klassen (*Product Level*) aufgelistet. Andere Satellitenbetreiber verwenden vergleichbare Einordnungen. In jedem Fall haben die Bilddaten, also die Pixelwerte, eine festgelegte (physikalische) Relevanz, die auch im Mosaik erhalten bleiben sollte.

Ein Ausgangsbild, also eine Luft- oder Satellitenaufnahme, stellt die Erdoberfläche perspektivisch verzerrt dar. Den Vorgang, eine solche Aufnahme zu entzerren und entsprechend einem gewählten Geländekoordinatensystem zu projizieren, wird *Orthorektifizieren* genannt. Die Projektion erfolgt auf Grundlage eines Modells der Abbildungseigenschaften des Aufnahmesystems sowie bekannter Positions- und Lageparameter und berücksichtigt dabei auch die Topografie der abgebildeten Oberfläche (Höhenmodell). Üblicherweise werden auch genau vermessene Passpunkte (*Ground Control Points*) einbezogen um eine hohe Genauigkeit zu erreichen [4].

Als Folge der Orthorektifizierung ergibt sich meistens ein trapezförmiger Definitionsbereich einzelner Orthofotos mit, im Allgemeinen, komplexem Randverlauf anstelle gerader Linien (Abbildung 1.1). Die digitale Speicherung der Orthofotos erfolgt am DLR

Product Level	Definition
0	Raw data after restoration of the chronological data sequence for the instrument(s) operating in observation mode, at full space/time resolution with all supplementary information to be used in subsequent processing (e.g. orbital data, health, time conversion, etc.) appended. Level 0 data are time-tagged. The precision and accuracy of the time-tag shall be such that the measurement data may be localized to accuracy compatible with the users' requirements.
1A	Level 0 data with corresponding radiometric, spectral and geometric (i.e. Earth location) correction and calibration computed and appended, but not applied.
1B	Level 1a data not re-sampled, quality-controlled and radiometrically calibrated, spectrally characterised, geometrically characterised, annotated with satellite position and pointing, landmarks and preliminary pixel classification (e.g. land/water/cloud mask). The Level 1b product consists of Top of Atmosphere (TOA) radiance ( $W m^{-2} sr^{-1} \mu m^{-1}$ ).
1C	Level 1B data orthorectified, re-sampled to a specified grid. Image re-sampling shall be performed using a selectable re-sampling method including at least bi-cubic convolution interpolation and nearest neighbour [...].
2A	Earth located pixel values converted to ground surface reflectance, i.e. after atmospheric corrections.
2C	Earth located pixel values converted to geophysical variables (land cover or geo-biophysical variable maps).
3	Spatially and/or temporally re-sampled biophysical variables and land cover maps.

**Tabelle 1.1:** Von der Europäischen Weltraumorganisation ESA für die Sentinel-2-Mission spezifizierte Product Level [3]

in einem proprietären *Xdibias*-Bildformat [57] oder als GeoTIFF-Datei und grundsätzlich zusammen mit Metadaten zur geografischen Lage sowie weiteren Informationen über die Bilddaten. Der gespeicherte Bildausschnitt entspricht aus technischen Gründen dem minimal umgebenden Rechteck des nach Norden ausgerichteten Definitionsbereichs. Das *Xdibias*-Format verwendet einen ausgezeichneten Pixelwert zur Kennzeichnung ungültiger Bildbereiche. In dieser Arbeit werden die Begriffe *Orthofoto* und *Szene* synonym benutzt. Sie beziehen sich auf die gültigen Bildbereiche und nicht auf den tatsächlich gespeicherten, rechteckigen Bildausschnitt.

### 1.2.2 Anforderungen und Ziele

Diese Arbeit strebt an, ein Verfahren zu konzipieren und zu implementieren, das ein ausgedehntes Orthofotomosaik aus teilweise überlappenden Einzelaufnahmen zusammensetzt. Entsprechend der Produktklasse des zu mosaikierenden Datensatzes, haben die Pixelwerte eine naturwissenschaftliche Bedeutung, die es zu erhalten gilt. Ähnliches gilt für die abgebildete Boden- beziehungsweise Bildtextur. Um dem Rechnung zu tragen, sollen Bereiche des Mosaiks jeweils *eindeutig* einer Szene zugeordnet werden. Umgekehrt ist es explizit nicht erwünscht, dass sich der Wert eines Pixels im finalen Mosaik, aus Anteilen verschiedener Szenen zusammensetzt, was auch ein Überblenden zwischen den Beiträgen verschiedener Szenen ausschließt. Ob sich die Unterteilungen, die sich aus der Anwendung einer bestimmten Methode ergeben prinzipiell für ein nachträgliches Überblenden eignen, sollte dennoch getestet werden. Ein solches Verfahren wäre dann universell einsetzbar.

Formal stellt sich die Situation wie folgt dar: Der zu mosaikierende Datensatz bestehe aus den  $N$  Szenen  $S = \{1, 2, \dots, N\}$  mit zugehörigem Definitionsbereich  $\Sigma_i$ . Der Definitionsbereich des Mosaiks  $\Sigma^{fin}$  ergibt sich dann gemäß

$$\Sigma^{fin} = \bigcup_{i \in S} \Sigma_i \quad . \quad (1.1)$$

Angestrebt wird ein Verfahren, das für jedes Orthofoto einen disjunkten Beitrag  $\Sigma_i^{fin}$  zum finalen Mosaik bestimmt. Für die Beiträge muss demnach gelten:

$$\forall j \neq i : \Sigma_i^{fin} \cap \Sigma_j^{fin} = \emptyset \quad (1.2)$$

$$\Sigma^{fin} = \bigcup_{i \in S} \Sigma_i^{fin} \quad (1.3)$$

Die Beiträge zweier Szenen sind genau dann benachbart, wenn ein gemeinsamer Rand  $\xi_{i,j} \neq \emptyset$  existiert.  $\xi_{i,j}$  wird als Nahtlinie oder Seamline bezeichnet und ist definiert als

$$\xi_{i,j} = \partial \Sigma_i \cap \partial \Sigma_j \quad . \quad (1.4)$$

Entlang einer solchen Nahtlinie kann im Mosaik eine künstliche Kante entstehen, also eine Kante, die in den Ausgangsszenen nicht oder nur schwach vorhanden ist. Als Kante wird hierbei eine Abgrenzung flächiger Bildelemente (homogene Textur, Tonwert,...) verstanden. Es gilt nun die Beiträge zum Mosaik derart zu bestimmen, dass keine künstlichen Kanten entstehen oder, sollte dies nicht möglich sein, die Nahtlinien geeigneten Geländestrukturen folgen. Dabei gilt es folgende Anforderungen zu berücksichtigen:

**Iteratives Mosaikieren:** Die zu mosaikierenden Datensätze sind im Allgemeinen nicht statisch und werden von Zeit zu Zeit aktualisiert. Szenen können dabei sowohl hinzugefügt, als auch entfernt werden. Es sollte möglich sein, ein bestehendes Mosaik inkrementell zu aktualisieren, ohne die gesamte Berechnung zu wiederholen. Die Reihenfolge der Änderungen sollte dabei keinen Einfluss auf das resultierende Mosaik haben. Ein in mehreren Inkrementen berechnetes Mosaik soll demnach dem Ergebnis der Berechnung in nur einem Schritt entsprechen.

**Inhomogene Radiometrie:** In dieser Arbeit werden Ausgangsbilder entsprechend Produktklasse 1C (Tabelle 1.1) oder besser vorausgesetzt. Dennoch ist aufgrund vielfältiger Effekte mit Helligkeits- und Kontrastabweichungen oder allgemein abweichender Radiometrie (hier: Intensitätsverteilung) der zu mosaikierenden Szenen zu rechnen. Ursachen sind etwa unterschiedliche Aufnahmezeiten (insbesondere Jahreszeiten, da die Tageszeit wegen des üblicherweise genutzten sonnensynchronen Orbits nicht variiert), atmosphärische Effekte und der Schwenk- beziehungsweise Blickwinkel des Sensorsystems (Abhängigkeit des Reflexionsverhaltens einer Oberfläche vom Einfallswinkel: Bidirektionale Reflektanzverteilungsfunktion). Auch kann es wünschenswert sein, Aufnahmen verschiedener Sensoren in einem Mosaik zu vereinen, wobei die spektrale Auflösung der Kanäle im Allgemeinen nicht identisch ist. Aus den genannten Gründen soll nicht davon ausgegangen werden, dass die Radiometrie in unterschiedlichen Aufnahmen für identisch gewählte Bildausschnitte übereinstimmt. Entsprechende Unterschiede sollen toleriert werden (Robustheit).

**Inhomogene Geometrie:** Die Georeferenzierung der zu verarbeitenden Orthofotos ist gewöhnlich auf den Subpixelbereich genau und die Eingabebilder sind entsprechend einem gemeinsamen Pixelraster ausgerichtet. Die geometrische Konsistenz überlappender Orthofotos ist ebenso sehr hoch, die Verschiebung identischer Merkmale zueinander ist also minimal. Im Allgemeinen soll jedoch davon ausgegangen werden, dass die Geometrie von Szenen, die sich überlappen, Unterschiede zeigt.

**Unterschiedliche Bildinhalte:** Ein mögliches Verfahren soll gegenüber Unterschieden in der Geländestruktur, Bodentextur, Vegetation,... oder allgemein abweichenden Bildinhalte verschiedener Orthofotos im gemeinsamen Überlappungsbereich robust sein. Dies ist notwendig, da die Aufnahmen eines Datensatzes häufig in verschiedenen Jahren beziehungsweise Jahreszeiten aufgenommen wurden.

**Unerwünschten Strukturen:** Gegebenenfalls werden in einem manuellen oder automatischen Vorverarbeitungsschritt unerwünschte Bildbereiche bestimmt. Dies kann zum Beispiel der Reduktion der Wolkenbedeckung im finalen Mosaik dienen. Insofern darf der Mosaikierungsalgorithmus keine Anforderungen an die Form des Definitionsbereichs der einzelnen Szenen stellen.

**Datenvolumen:** Die geometrische Auflösung gängiger Aufnahmesysteme reicht in den Meterbereich. Der Tonwertumfang jedes Kanals beträgt bis zu 16 Bit. Typisch für Multispektralaufnahmen ist die Größenordnung von 200 Megapixeln bei vier Kanälen. Realistisch sind über ein Gigabyte pro Szene. Ein Verfahren, das für die Mosaikierung ausgedehnter Gebiete ( $\gg 100$  Szenen) geeignet ist, muss deshalb auf die Verarbeitung großer Datenmengen ausgerichtet sein. Dementsprechend sollte es möglichst linear skalieren. Desweiteren kann nicht vorausgesetzt werden, dass das gesamte Mosaik im Hauptspeicher vorgehalten werden kann.

**Software:** Die zu nutzende Softwareumgebung wird mit dem Linux-Betriebssystem *CentOS 5* [55] betrieben. Es stehen *Python 2.4* [516], *GCC 4.1.2* [511] und *Xdibias* [57] zur Verfügung. Die Orthofotos liegen in einem proprietären *Xdibias*-Format vor. Um auch Bibliotheken und Software evaluieren zu können, die unter *CentOS 5* nicht zur Verfügung steht, wurde ein aktuelles *Gentoo*-Linux [512] in einer Prefix-Umgebung (siehe [512]) eingerichtet. Die Kernfunktionalität der Implementierung sollte allerdings unter *CentOS 5* nutzbar sein.

### 1.3 Verwandte Arbeiten

Das vollautomatische Mosaikieren von Fernerkundungsdaten und dessen Aspekte sind das Thema einer Vielzahl an Arbeiten. Die einzelnen Arbeitsschritte sind dabei häufig von interdisziplinärer Bedeutung - ganz besonders in der Informatik, aber auch in Naturwissenschaften wie der Biologie (Mikroskopie) und weiteren Disziplinen. Eine große Auswahl relevanter Publikationen findet sich etwa im Umfeld der Schlagworte *Image Stitching* und *Panorama Stitching*. Einen guten Überblick gibt „*Image Alignment and Stitching: A Tutorial*“ von Szeliski (2006) [5].

Häufig wird das Überblenden zwischen den Beiträgen verschiedener Bilder angestrebt. Die Bildung eines gewichteten Mittelwerts der Pixelwerte (*Feathering*) ist häufig keine gute Lösung [6, Kap. 9]. Bewährt hat sich hingegen ein Verfahren, das von Burt und Adelson in den frühen 80er Jahren unter dem Titel „*A Multiresolution Spline With Application to Image Mosaics*“ [7] publiziert wurde. Charakteristisch ist das Überblenden in Abhängigkeit von der Ausdehnung beziehungsweise der Ortsfrequenz der Bildmerkmale. Dies geschieht durch separates Mischen der Frequenzbänder einer Laplace Pyramide. Beeindruckende Ergebnisse lassen sich durch Operationen im Gradientenraum erreichen: Pérez, Gangnet und Blake bestimmen in „*Poisson Image Editing*“ (2003) [8] eine Interpolationsfunktion

durch Lösen der Poisson-Differenzialgleichung, welche die ursprünglichen Gradienten bestmöglich beibehält. Das Verfahren ist effizient anwendbar (Szeliski, Uyteendaele und Steedly - „*Fast Poisson Blending using Multi-Splines*“, 2011) [9].

Im Vorfeld müssen gewöhnlich Nahtlinien bestimmt werden, an denen sich das *Blending*-Verfahren orientiert. Eine gut gewählte Naht erzeugt keine künstlichen Artefakte und ist im besten Fall auch ohne Überblenden nicht wahrnehmbar. Die Wahl des Nahtverlaufs erfolgt bei den meisten Verfahren auf Basis einer Kostenfunktion unter Berücksichtigung (lokaler) radiometrischer oder geometrischer Unterschiede zwischen den überlappenden Ursprungsbildern. In jedem Fall ist die Suche nach geeigneten Nahtlinien equivalent zur Unterteilung des Mosaiks in disjunkte Beiträge.

Einen wichtigen Beitrag liefert Veröffentlichung „*Interactive Digital Photomontage*“ (Agarwala, Dontcheva, Agrawala et. al) von 2004 [10]. Mit einem Graph-Cut basierten Algorithmus wird die Zuordnung der Pixel zu den Ursprungsbildern mit den niedrigsten Gesamtkosten (approximativ) bestimmt. Gracias, Gleason, Negahdaripour et. al („*Fast Image Blending using Watersheds and Graph Cuts*“, 2009) [11] verwenden ein solches Graph-Cut-Labeling zur Generierung eines Unterwassermosaiks. Aus Geschwindigkeitsgründen führen sie eine Vorsegmentierung der Überlappungsbereiche mit einer Wasserscheidentransformation durch und reduzieren das Problem auf binäres Labeling. Prados, Garcia, Racias et. al präsentieren in „*A Novel Blending Technique for Underwater Gigamosaicing*“ (2012) [12] ein umfassendes Graph-Cut basiertes Verfahren für Unterwasseraufnahmen. Insbesondere geben die Autoren auch einen umfangreichen Überblick über 20 verschiedene Verfahren zur Mosaikierung.

Darüber hinaus gibt es eine Vielzahl an Veröffentlichungen aus dem Bereich der Erdfernerkundung, die sich mit der Erzeugung eines Orthofotomosaiks und dem Suchen geeigneter Nahtlinien im Speziellen beschäftigen. Fernandez, Garfinkel und Arbiol erörtern in „*Mosaicking of Aerial Photographic Maps via Seams Defined by Bottleneck Shortest Paths*“ (1998) [13] die Verwendung eines Minimax-Pfad-Algorithmus<sup>1</sup> auf Basis eines Bisektionsverfahrens. Die beschriebene Methode ist Bestandteil der Prozessierungskette des Institut Cartografic de Catalunya (ICC) und ist auf Realdaten erprobt. Eine wichtige Referenz ist „*Mosaicking of Orthorectified Aerial Images*“ (Afek und Brand, 1998) [17]. Afek und Brand bestimmen die Nahtlinie in einem mehrstufigen Verfahren auf Basis von Verknüpfungspunkten (*Tie Points*) und einer Variante des Dijkstra-Algorithmus. Kerschner beschreibt in „*Seamline Detection in Colour Orthoimage Mosaicking by Use of Twin Snakes*“ [2] (2001) ein Verfahren auf Grundlage aktiver Konturen (*Snakes*). Yu, Holden, Dentith et. al („*Towards the Automatic Selection of Optimal Seam Line Locations When Merging Remote-Sensing Images*“, 2012) [18] berücksichtigen ein Salienz-Kriterium auf Basis

<sup>1</sup>Die Autoren verwenden den Ausdruck *Bottleneck Shortest Path*. Diese Bezeichnung ist nicht klar definiert: Häufig wird unter einem *Bottleneck (Shortest) Path Problem* die Bestimmung eines Weges mit maximaler Kapazität zwischen festgelegten Start- und Endknoten (Maximin-Problem) verstanden [14], [15]. Die Autoren beziehen sich jedoch auf das Finden eines Pfades, dessen maximales Kantengewicht minimal ist (Minimax-Pfad). Auch dies wird in der Literatur als *Bottleneck Shortest Path Problem* bezeichnet [16].

eines *Human Attention Model* (der Begriff Salienz wird in Fußnote 1 auf Seite 15 erläutert). Zur Bestimmung der paarweisen Nahtlinie wird ein Verfahren aus dem Bereich der dynamischen Programmierung eingesetzt. Chon, Kim und Lin schränken in „*Seam-Line Determination for Image Mosaicking: A Technique Minimizing the Maximum Local Mismatch and the Global Cost*“ (2010) [19] den Suchraum zunächst auf Bereiche ein, in denen mögliche Minimax-Pfade verlaufen können. Der Dijkstra-Algorithmus wird schließlich zur Bestimmung der Nahtlinie verwendet. Die Kostenfunktion basiert auf der Korrelation der Pixelwerte zwischen den Ursprungsbildern. Bei der Mosaikierung von Pleiades-Daten verwendet die französische Raumfahrtagentur Centre national d'études spatiales (CNES) ein Verfahren, das auf einer vorangegangenen Veröffentlichung von Chon und Lin [20] basiert (Bignalet-Cazalet, Baillarin und Panem, „*Automatic and Generic Mosaicing of Multisensor Images: An Application To Pleiades HR*“, 2012) [21].

Die im vorigen Absatz genannten Arbeiten betrachten Nahtlinien zwischen Paaren überlappender Orthofotos. Soille stellt in „*Morphological Image Compositing*“ (2006) [22] ein Verfahren vor, das in der Lage ist, auch Schnittbereiche von mehr als zwei Aufnahmen reihenfolgenunabhängig zu bearbeiten. In sich überdeckenden Aufnahmen werden dazu gemeinsame, auffällige Bildmerkmale bestimmt. Die Platzierung der Nahtlinie erfolgt mit einer Variante der Wasserscheidentransformation. Die Methode wurde am Joint Research Center (JRC) der EU-Kommission erfolgreich zur Mosaikierung des IMAGE-2006 Referenzdatensatzes der ESA eingesetzt [23]. Wan, Wang, Xiao et. al („*Automatic Determination of Seamlines for Aerial Image Mosaicking Based on Vector Roads Alone*“, 2013) [24] nutzen zusätzliche Kartendaten (unter anderem Straßenzüge) anstatt der Bildinhalte zur präzisen und effizienten Mosaikierung großer Datensätze.

Bei der Produktion eines Orthofotomosaiks ist gegebenenfalls eine manuelle Qualitätskontrolle erwünscht. Dabei kommen bei der Platzierung der Nahtlinie häufig mehrere, semantisch verschiedene Möglichkeiten in Frage. Summa, Tierny und Pascucci stellen in „*Panorama Weaving: Fast and Flexible Seam Processing*“ (2012) [25] eine Methode zur automatischen Wahl der Seamlines vor, die eine effiziente interaktive Nachbearbeitung ermöglicht. Bei Unterschieden in den Bildinhalten wird dadurch die manuelle Berücksichtigung semantisch verschiedener Lösungen möglich.

Gerade bei der Verwendung von Orthofotomosaiken als Kartenhintergrund in GIS-Systemen (Geoinformationssystem) müssen unterschiedliche Skalierungen berücksichtigt und dabei verschiedene Datensätze integriert werden. Die Skala kann als dritte Dimension des Mosaiks angesehen werden. Han und Hoppe erläutern in „*Optimizing Continuity in Multiscale Imagery*“ (2010) [26] die Erzeugung einer kontinuierlichen Bildpyramide aus Satelliten- und Luftbildaufnahmen.



## 1.4 Aufbau der Arbeit

Schwerpunkt dieser Arbeit sind Methoden zur geeigneten Platzierung von Nahtlinien im Überlappungsbereich mehrerer Orthofotos. Solche Nahtlinien grenzen dabei Bildregionen voneinander ab, die unterschiedlichen Einzelaufnahmen entnommen werden. Anders ausgedrückt wird jedem Pixel des Mosaiks ein eindeutiges Ursprungsbild zugewiesen.

Kapitel 2 betrachtet zwei einfache Verfahren, die ohne viel Aufwand umgesetzt werden können und unter gewissen Voraussetzungen gute Ergebnisse liefern. Allerdings sind diese Methoden nur eingeschränkt mit den genannten Anforderungen (Abschnitt 1.2) zu vereinbaren.

Im weiteren Verlauf der Arbeit beschränkt sich die Betrachtung auf Verfahren, deren Ergebnis nicht von der Reihenfolge abhängt, in der die Ausgangsbilder geladen beziehungsweise bearbeitet werden. Die geschilderten Methoden gehen somit über die Betrachtung von paarweisen Nahtlinien hinaus. Auch wurde Wert darauf gelegt, dass die Algorithmen die inkrementelle Mosaikierung unterstützen oder dahingehend modifiziert werden können.

In Kapitel 3 wird das Mosaikieren zunächst als Optimierungsproblem aufgefasst. Der sich ergebende Energieterm wird dabei mit speziellen *Graph-Cut*-Methoden minimiert. Im Zuge dessen wird ersichtlich, dass sich der betrachtete Algorithmus wegen des globalen Charakters und der hohen Laufzeitkomplexität nicht direkt für die Erzeugung ausgedehnter Mosaiken eignet. Deshalb werden Möglichkeiten skizziert, die Komplexität zu reduzieren und eine globale Lösung durch lokale Berechnungen zu approximieren. Letztlich ermöglicht dies auch das iterative Erweitern eines bestehenden Mosaiks.

Kapitel 4 beschreibt das sogenannte *Morphological Image Compositing*, das Nahtlinien an salienten<sup>1</sup> Bildstrukturen ausrichtet. Dieses Verfahren betrachtet Schnittbereiche beliebigen Grades und funktioniert reihenfolgenunabhängig. Die zugrundeliegende Idee ist für das iterative Mosaikieren geeignet, was dann allerdings Anpassungen des beschriebenen Verfahrens erfordert.

Aufbauend auf *Morphological Image Compositing* wird in Kapitel 5 ein angepasstes Verfahren beschrieben, das die Verwendung von Rasterdaten zur Repräsentation von Zwischenschritten zugunsten von Vektordaten einschränkt. Dadurch wird das iterative Mosaikieren auf einfache Weise möglich und es ergeben sich positive Effekte auf den Speicherbe-

---

<sup>1</sup>Der Begriff *Salienz* beziehungsweise das abgeleitete Adjektiv *salient* wird im nachfolgenden Text zur Bezeichnung von visuell hervorstechenden Merkmalen verwendet. Der Begriff *Salienz* (im Sinne von Auffälligkeit oder Deutlichkeit) entstammt psychologischer Fachliteratur und wird insbesondere im Bereich der Wahrnehmungs- und Aufmerksamkeitsforschung gebraucht. Siehe dazu etwa [27].

darf und die Skalierbarkeit des Verfahrens. Als Nebeneffekt können zusätzlich Artefakte gefiltert werden, die bei dem ursprünglichen Verfahren auftreten können. Darüberhinaus wurde der Gesamtablauf von *MSeam* flexibel und erweiterbar gestaltet. In Kapitel 6 werden mit diesem Verfahren erzielte Ergebnisse betrachtet.

Im Rahmen der Arbeit ist eine erweiterbare Softwareumgebung („*FAMOS - Fully Automated Mosaicking Study*“) entstanden, die auf die Verwendung von Vektordaten ausgerichtet ist. Darauf aufbauend wurde eine *Strategie* für die modifizierte Variante des *Morphological Image Compositing* implementiert und getestet (*MSeam*). Unterstützt werden auch die beiden einfachen Algorithmen aus Kapitel 2 und das in Kapitel 3 beschriebene *Pixel Labeling*.

# Einfache Verfahren

---

Häufig lassen sich mit einer ausgesprochen einfachen Herangehensweise bereits gute Ergebnisse erzielen. Als naheliegender Ansatz wird das simple Übereinanderlegen der zu mosaikierenden Szenen betrachtet. Außerdem wird auf die Zuordnung von Pixeln zur nächstgelegenen Szene eingegangen. Dieses Vorgehen wird nachfolgend als Voronoisegmentierung bezeichnet.

## 2.1 Einfaches Übereinanderlegen

Das als Erstes betrachtete Verfahren erfordert keine besondere Erklärung: Disjunkte Beiträge der Ausgangsbilder werden durch einfaches Übereinanderlegen der Szenen bestimmt. Der Beitrag eines Orthofotos  $\Sigma_i^{fin}$  zum finalen Mosaik bestimmt sich demnach aus dem Definitionsbereich  $\Sigma_i$  der ursprünglichen Szene abzüglich der Definitionsbereiche  $\Sigma_{j>i}$  nachfolgend eingefügter Fotos:

$$\Sigma_i^{fin} = \Sigma_i \setminus \bigcup_{j>i} \Sigma_j \quad (2.1)$$

Das Ergebnis ist offensichtlich abhängig von der Reihenfolge in der die Bilder dem Mosaik hinzugefügt werden. Die Nahtlinien zwischen den Beiträgen unterschiedlicher Szenen folgen dabei jeweils der Umrandung mindestens eines Orthofotos, ein Kaschieren der Seamline durch (symmetrisches) Überblenden ist somit nicht ohne weiteres anwendbar.

Alternativ können die Tonwerte aller Szenen, die einen bestimmten Bereich überdecken, punktwise und im gesamten Überlappungsbereich miteinander verrechnet werden. Das Ergebnis ist dann unabhängig von der Reihenfolge, in der die Bilder in das Mosaik eingefügt werden, wobei jedoch das Resultat nicht der eigentlich angestrebten Aufteilung des Mosaiks in disjunkte Beiträge entspricht. Es bietet sich insbesondere die Bestimmung des Medians an, womit Ausreißer in einigen der Szenen, wie zum Beispiel Wolken, vom Mosaik ausgeschlossen werden können. Der Median entspricht auch wieder einer eindeutigen Zuordnung jedes Pixels zu einer Einzelaufnahme.

In der Praxis wird das einfache Übereinanderlegen von Szenen zur Echtzeitvisualisierung mehrerer Orthofotos ohne Vorberechnung eines aufwändigeren Mosaiks eingesetzt.

Beispiele für eine solche Nutzung sind etwa das quelloffene GIS-System *QGIS* [S18] und die Software *OpenEV* [S3] - ein Betrachter für georeferenzierte Bilddaten.

## 2.2 Voronoisegmentierung

Die Methode geht von Bildbereichen aus, die von vornherein einem Ursprungsbild eindeutig zugeordnet werden können, in denen sich also keine Szenen überlappen. Diese werden im Zuge der Voronoisegmentierung auf angrenzende Überlappungsbereiche ausgedehnt. Dazu werden alle Bildpunkte, die in mehreren Orthofotos enthalten sind, derjenigen Szenen zugeordnet, die das nächstgelegene, eindeutig bestimmte Pixel beiträgt.

Der Definitionsbereich der Szenen sei bei der folgenden Betrachtung wieder  $\Sigma_i$ . Desweiteren sei  $L(p)$  die Menge der an einem Bildpunkt  $p$  definierten Orthofotos,  $L^{fin}(p)$  die Szene, die das Pixel  $p$  zum finalen Mosaik beisteuert (Beitrag ist wieder  $\Sigma_i^{fin}$ ) und  $NN(p)$  die Menge der nächstgelegenen Szenen.

$$L(p) = \{i \mid p \in \Sigma_i\} \quad (2.2)$$

$$NN(p) = \{i \in L(p) \mid \forall j \in L(p) : dist(\Sigma_i, p) \leq dist(\Sigma_j, p)\} \quad (2.3)$$

$$L^{fin}(p) = \begin{cases} NN(p) & \text{wenn } \|NN(p)\| = 1 \\ \text{unbestimmt} & \text{sonst } (\|NN(p)\| > 1) \end{cases} \quad (2.4)$$

$$\Sigma_i^{fin} = \{p \in \Sigma_i \mid L^{fin}(p) = i\} \quad (2.5)$$

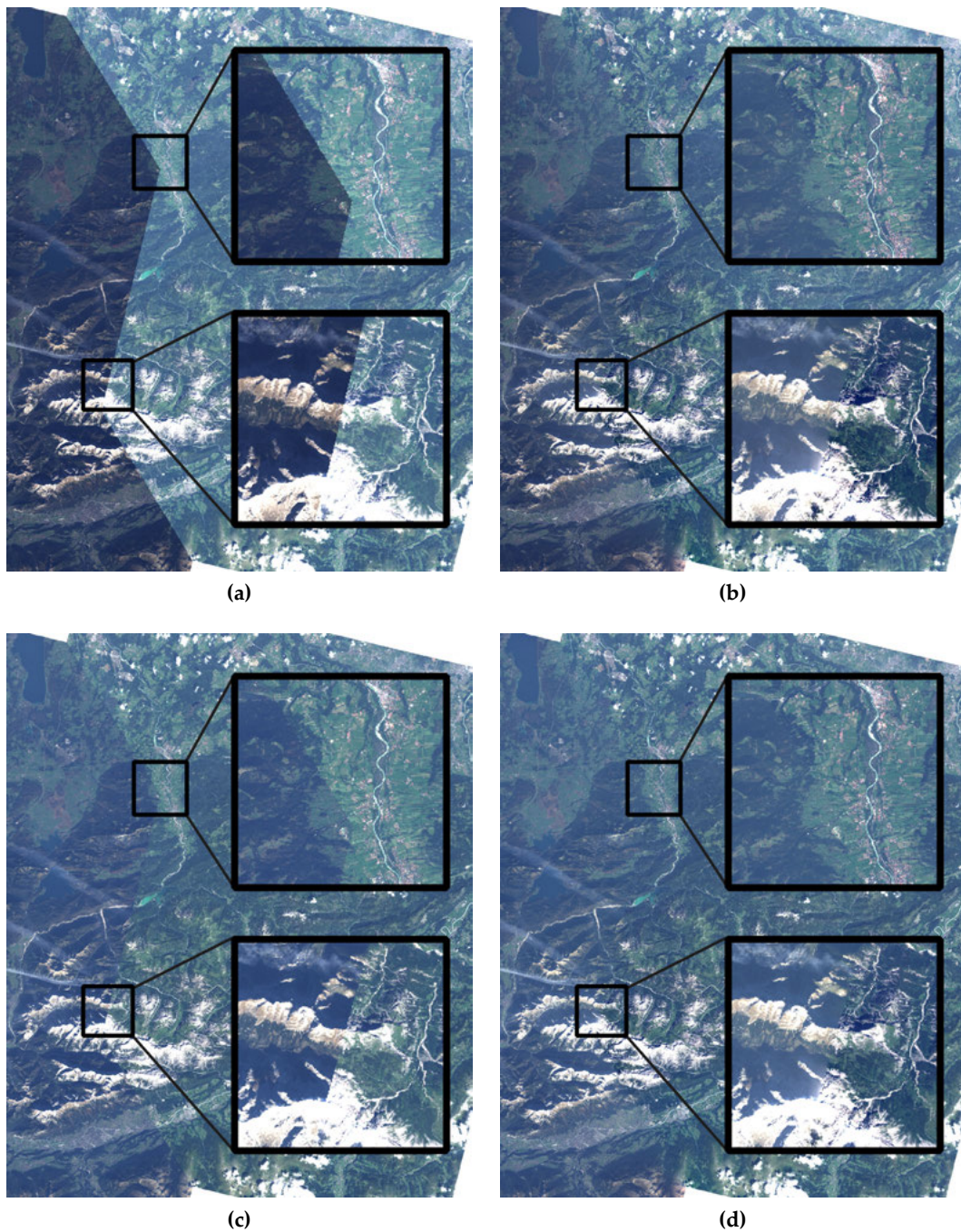
$dist(\sigma, p)$  ist die Länge des kürzesten, gänzlich innerhalb von  $\sigma$  verlaufenden Weges, von  $p$  zum nächstgelegenen, eindeutig bestimmten Bildpunkt

$$p' \in \sigma \setminus \bigcup_{i \in L(p) \setminus \sigma} \Sigma_i \quad (2.6)$$

Die Einschränkung auf Pfade innerhalb von  $\sigma$  berücksichtigt die prinzipiell beliebige Form der Definitionsbereiche: Der Benutzer kann ungültige Ausschnitte uneingeschränkt maskieren.

Es kann dabei Punkte innerhalb des Mosaiks geben, für die mehr als eine nächstgelegene Szene existiert. In diesem Fall muss die Bestimmung des beitragenden Orthofotos nach weiteren Kriterien erfolgen. Auf Grundlage der Voronoisegmentierung ist eine iterative Erweiterung des Mosaiks möglich, wobei nur Bildbereiche aktualisiert werden müssen, die von den hinzuzufügenden Szenen überdeckt werden.

Die Beispiele in Abbildung 2.1 wurden mit diesem Verfahren erzeugt. Insbesondere im Fall nicht radiometrisch angeglicher Szenen sind die entstehenden geraden Nahtlinien deutlich zu erkennen. Wird jedoch zusätzlich zwischen den Beiträgen benachbarter



**Abbildung 2.1:** Ausschnitt eines Mosaiks aus zwei Orthofotos (Sensor: ALOS AVNIR-2 [28]). Abweichende Aufnahmezeitpunkte (2006-10-27 und 2007-07-13) sorgen für sichtbare Unterschiede zwischen den Bildern. In (c) und (d) wurde zunächst eine radiometrische Angleichung durchgeführt. (a) und (c) zeigen das Ergebnis der Voronoisegmentierung; in (b) und (d) wurde zusätzlich zwischen den Beiträgen der beiden Szenen überblendet. Die Satellitenaufnahmen zeigen die Gegend um Innsbruck, Österreich (unten links im Bild).

Szenen überblendet, erhält man gute Resultate. Bei vorangegangener radiometrischer Angleichung der Ausgangsbilder wird die Nahtlinie fast vollständig kaschiert. In Bereichen mit Unterschieden in den Bildinhalten lassen sich Artefakte aber nur schwer vermeiden.

Für die Berechnung der Mosaiken wurde die im Rahmen dieser Arbeit entstandene Software *FAMOS* genutzt. Die Voronoisegmentierung ergibt sich als Sonderfall der in Abschnitt 5 beschriebenen Methode zur Bestimmung von Nahtlinien. Für das Überblenden der Szenen wurde *Multiblend* [56] verwendet. Die radiometrische Angleichung erfolgte mit *Xdibias* [57].

### Pixel Labeling - Das optimale Mosaik?

---

Die unterschiedlichsten technischen und naturwissenschaftlichen Fragestellungen lassen sich als Optimierungsproblem auffassen. Das bedeutet, dass für ein gegebenes Problem, eine Energie- beziehungsweise Kostenfunktion aufgestellt werden kann. Gesucht ist dann eine Lösung mit minimalen Kosten. Entsprechend lässt sich auch ein Mosaik, das aus disjunkten Beiträgen mehrerer Szenen zusammengesetzt werden soll, bewerten. Die nachfolgende Betrachtung führt zu einem Graphalgorithmus. Das Mosaik wird dabei als Graph dargestellt, wobei jeder Knoten des Graphen einem oder mehreren Pixeln des Mosaiks entspricht. Gesucht ist nun eine eindeutige Zuordnung (Kennzeichnung) der Knoten zu den Ausgangsszenen (*Pixel Labeling*), die bezüglich einer zuvor festgelegten Kostenfunktion optimal ist.

Im Ergebnis setzt sich das Mosaik aus Beiträgen verschiedener Szenen zusammen. Bei der Beurteilung eines solchen Mosaiks gilt es mehrere Kriterien zu berücksichtigen. Von wesentlicher Bedeutung sind dabei die entstandenen Nahtlinien (Abschnitt 1.2), die möglichst unscheinbar verlaufen sollten. Eine Nahtlinie ergibt sich, wenn benachbarte Pixel verschiedenen Szenen zugeordnet werden. Entsprechend muss eine mögliche Kostenfunktion jeweils Paare benachbarter Pixel berücksichtigen. Unter dieser Voraussetzung lassen sich die Gesamtkosten des Mosaiks als Summe je eines Einzelterms für jedes Paar berechnen. Diese Darstellung wird sich bei der Bestimmung einer optimalen Lösung als zweckmäßig erweisen.

Nun gibt es  $k^n$  Möglichkeiten, ein Mosaik mit  $n$  Pixeln aus den Beiträgen von  $k$  Szenen zu bilden. Wegen der exponentiell vielen Varianten ist es im Allgemeinen nicht möglich, alle denkbaren Varianten aufzustellen, einzeln zu bewerten und dann dasjenige, mit den geringsten Kosten auszuwählen. Es ist dennoch möglich, ein Mosaik mit minimalen Kosten in polynomieller Zeit zu bestimmen. Dabei wird ausgenutzt, dass der Kostenbeitrag einer einzelnen Pixelzuordnung zu den Gesamtkosten nur von dessen direkter Umgebung abhängt. In der Vergangenheit hat diesbezüglich eine Klasse von Graphalgorithmen Bekanntheit erlangt, die derartige kombinatorische Optimierungsprobleme auf Schnitte in Graphen (*Graph Cuts*) zurückführt [29]–[32].

### 3.1 Das Mosaik als Graph

Zunächst muss das Mosaik als Graph repräsentiert werden. Dabei wird folgende Begrifflichkeit verwendet:

**Knoten** Sei  $\mathcal{V}$  eine Menge von Knoten.

**Kanten** Knoten können jeweils paarweise durch eine Kante  $e$  verbunden sein. Die Menge der Kanten sei  $\mathcal{E}$ . Dabei existiert nicht notwendigerweise für jedes Paar von Knoten eine Kante, es gilt also

$$\mathcal{E} \subseteq \{ \{v_i, v_j\} \mid v_i, v_j \in \mathcal{V} \} \quad . \quad (3.1)$$

Einer Kante kann ein Gewicht  $w_e$  zugewiesen werden. Im Rahmen der nachfolgenden Betrachtung muss gelten  $w_e > 0$ .

**Graph**  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  wird als Graph bezeichnet. Wurden den Kanten Gewichte zugewiesen, spricht man von einem gewichteten Graphen.

**Weg** Eine Knotenfolge  $\omega = (v_1, v_2, \dots, v_n)$  mit  $\{v_i, v_{i+1}\} \in \mathcal{E}$  heißt Weg.

**Zusammenhängender Graph** Ein Graph ist zusammenhängend, wenn für jedes Paar  $(v_1, v_2) \in \mathcal{V} \times \mathcal{V}$  ein Weg von  $v_1$  nach  $v_2$  existiert.

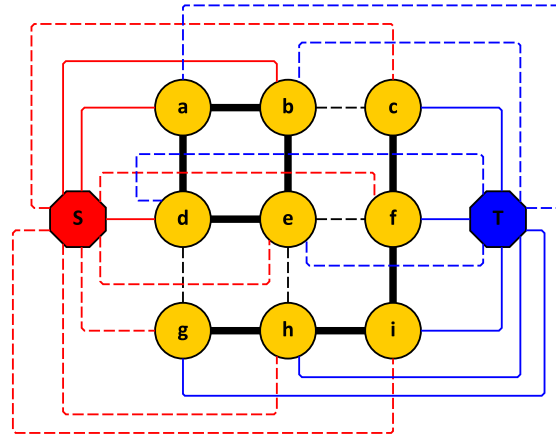
**Teilgraph**  $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$  ist ein Teilgraph von  $\mathcal{G}$  falls  $\mathcal{V}' \subset \mathcal{V}$  und  $\mathcal{E}' \subset \mathcal{E}$  mit  $v_1, v_2 \in \mathcal{V}'$  für  $\{v_1, v_2\} \in \mathcal{E}'$ .

**Schnitt** Ein Schnitt  $\mathcal{C}$  ist eine Menge von Kanten  $\mathcal{C} \subset \mathcal{E}$ , für die gilt, dass  $\mathcal{G}_S = (\mathcal{V}, \mathcal{E} \setminus \mathcal{C})$  aus zwei Teilgraphen besteht. In einem gewichteten Graph lassen sich die Kosten eines Schnitts als die Summe der Kantengewichte  $\|\mathcal{C}\| = \sum_{e \in \mathcal{C}} w_e$  definieren. Ein minimaler Schnitt  $\mathcal{C}^*$  ist ein Schnitt dessen Betrag gleich dem Minimum über alle Schnitte ist, so dass für alle weiteren Schnitte  $\mathcal{C}'$  gilt  $\|\mathcal{C}'\| \geq \|\mathcal{C}^*\|$ .

Ein Mosaik lässt sich dann direkt als Graph  $\mathcal{G}_M = (\mathcal{P}, \mathcal{N})$  darstellen: Jedes Pixel wird dazu als Knoten  $v_p \in \mathcal{P}$  aufgefasst, wobei ein solcher Knoten nachfolgend auch als *Pixelknoten* bezeichnet wird. Entsprechend der 4-Nachbarschaft des Pixelbilds werden die Pixelknoten durch Kanten miteinander verbunden (Kantenmenge  $\mathcal{N}$ ). Solche Kanten werden als *n-Links* bezeichnet.

Zusätzlich wird jede der Ausgangsszenen durch spezielle *Endknoten* repräsentiert, die bei Bedarf dem Graph hinzugefügt werden. Der resultierende Graph ist  $G'_M$ . Die Zuordnung eines Pixels des Mosaiks zu einer bestimmten Ausgangsszene wird in  $G'_M$  durch eine Kante zwischen Endknoten und jeweiligem Pixelknoten repräsentiert (als *t-Link* bezeichnet). Prinzipiell ist es erlaubt, einer Szene beliebige Pixelknoten zuzuordnen, auch wenn das jeweilige Pixel nicht im Definitionsbereich  $\Sigma_i$  (Abschnitt 1.2) der Szene enthalten ist. Eine solche Zuordnung ist lediglich temporär und wird wegen der damit verbundenen Kosten bei der finalen Aufteilung entfallen.





**Abbildung 3.1:** Beispiel eines ungerichteten Graph mit Source (S) und Target (T) als Endknoten. Ein möglicher Schnitt zur Aufteilung in zwei disjunkte Teilgraphen, die jeweils Source beziehungsweise Target zugeordnet sind, ist gestrichelt dargestellt.

### 3.2 Minimale Schnitte - Binäres Labeling

Grundlegend ist die Betrachtung des binären Labeling. Dabei werden dem Graph aller Pixelknoten zunächst zwei Endknoten (*Source S* und *Target T*) hinzugefügt. Im Zusammenhang mit der Mosaikierung repräsentieren diese zwei Szenen, aus denen ein Mosaik gebildet werden soll. Die grundlegende Problemstellung ist jedoch universell. In der Ausgangssituation sind die Endknoten mit allen Pixelknoten durch Kanten verbunden. Angestrebt wird nun eine geeignete Aufteilung  $S(\mathcal{C}) = \{\mathcal{G}_1, \mathcal{G}_2\}$  des Graphen  $\mathcal{G}'_M$  in zwei, in sich zusammenhängende Teilgraphen, die untereinander jedoch nicht verbunden sind. Dies geschieht durch Bestimmung eines geeigneten Schnitts  $\mathcal{C}$ . Abbildung 3.1 macht deutlich, dass für solche Schnitte zwei Eigenschaften gelten müssen:

1. Für alle  $p \in \mathcal{V}$  enthält  $\mathcal{C}$  genau einen der  $t$ -links  $\{p, S\}$  und  $\{p, T\}$ .
2.  $\mathcal{C}$  enthält die Kante  $\{p, q\}$  genau dann, wenn die Knoten  $p$  und  $q$  mit unterschiedlichen Endknoten verbunden sind.

Grundsätzlich gibt es eine Vielzahl möglicher Schnitte, die diese Eigenschaften erfüllen. Indem die Kanten des Graphen nun mit Gewichten versehen werden, lassen sich die Schnitte anhand der jeweiligen Kosten  $\|\mathcal{C}\|$  beurteilen. Die Gewichte werden dabei wie folgt festgelegt:

**Wechselwirkungspotential** (*Interaction Penalty*)  $V_{p,q}$  für alle Kanten  $p, q \in \mathcal{N}$ .

**Gebietspotential** (*Data Penalty*)  $G_p^{(t)}$  für alle Kanten  $\{p, t\}$  mit  $p \in \mathcal{P}, t \in \{S, T\}$ .

Für die Kosten eines Schnitts  $\mathcal{C}$  gilt somit

$$\|\mathcal{C}\| = \sum_{e \in \mathcal{C}} w_e = \sum_{\{p,q\} \in \mathcal{C} \cap \mathcal{N}} w_{\{p,q\}} + \sum_{\{p,S\} \in \mathcal{C}} w_{\{p,S\}} + \sum_{\{p,T\} \in \mathcal{C}} w_{\{p,T\}} \quad (3.2)$$

$$= \sum_{\{p,q\} \in \mathcal{C} \cap \mathcal{N}} V_{p,q} + \sum_{\{p,S\} \in \mathcal{C}} G_p^{(S)} + \sum_{\{p,T\} \in \mathcal{C}} G_p^{(T)} \quad (3.3)$$

$$= \sum_{\{p,q\} \in \mathcal{N}} (1 - \delta_{L(p),L(q)}) V_{p,q} + \sum_{p \in \mathcal{P}} G_p^{(L(p))} = K(S(\mathcal{C})) \quad (3.4)$$

mit  $K(S)$  den Kosten der Aufteilung (Segmentierung)  $S$ , der Kennzeichnung eines Knoten

$$L(p) = \begin{cases} S & p \text{ ist mit } S \text{ durch eine Kante verbunden} \\ T & p \text{ ist mit } T \text{ durch eine Kante verbunden} \end{cases} \quad (3.5)$$

und der  $\delta$ -Funktion

$$\delta_{L(p),L(q)} = \begin{cases} 1 & \text{wenn } L(p) = L(q) \\ 0 & \text{wenn } L(p) \neq L(q) \end{cases}. \quad (3.6)$$

Mit der Wahl der Kostenfunktion  $K(S)$  (3.4) für die Parametrisierung des ursprünglichen Optimierungsproblems, ist das Optimierungsproblem demnach auf die Suche eines minimalen Schnitts zurückgeführt. Ein minimaler Schnitt kann mit sogenannten *Max-Flow-Min-Cut*-Algorithmen (Abschnitt 3.3) in polynomieller Zeit bestimmt werden.

Die Kostenfunktion  $K(S)$  ist geeignet um die Unterteilung eines Mosaiks in disjunkte Beiträge der einzelnen Szenen zu bewerten:

1. Wurden zwei benachbarte Pixel der selben Szenen entnommen, verläuft zwischen diesen Pixeln keine Nahtlinie im Sinne von Abschnitt 1.2. Es ist demnach unproblematisch, dass die  $\delta$ -Funktion in Formel 3.4 in diesem Fall den paarweisen Wechselwirkungsterm unterdrückt. In allen anderen Fällen, also wenn zwei benachbarte Pixel unterschiedlichen Szenen zugeordnet wurden, ist eine Bewertung der Zuordnung durch geeignete Definition des Wechselwirkungspotentials möglich.
2. Der Gebietsterm beurteilt die Zuordnung eines Pixels zu einer bestimmten Szene ohne die Zuordnung benachbarter Pixel zu berücksichtigen. Dieses Gebietspotential kann genutzt werden, um die Kosten der Zuordnung eines Pixels  $p$  zu einer Szene  $i$  mit  $p \notin \Sigma_i$  (Abschnitt 1.2) mit unendlichen Kosten zu bewerten. Durch Summenbildung kann das Gebietspotential darüberhinaus mit weiteren Kriterien kombiniert werden. Beispiele werden in Abschnitt 3.5, Kostenfunktion, genannt.

### 3.3 Max-Flow Min-Cut

Es gibt etliche Standardverfahren um einen minimalen Schnitt in einem Graphen zu bestimmen. Diese Algorithmen arbeiten üblicherweise mit Flüssen und Netzwerken. Die verwendeten Definitionen sind:

**Gerichteter Graph** Ein Graph heißt gerichtet, wenn dieser keine ungerichteten Kanten  $\{p, q\}$  sondern stattdessen gerichtete Kanten  $(p, q)$  enthält.

**Netzwerk** Ein gewichteter, gerichteter, (schwach) zusammenhängender Graph  $\mathcal{G}_N = (\mathcal{V} \cup \{S, T\}, \mathcal{E}' \supset \mathcal{E})$  mit zwei ausgezeichneten Knoten  $S$  (Quelle) und  $T$  (Senke) wird als Netzwerk bezeichnet. Die Kantengewichte  $w_e$  eines Netzwerks heißen Kapazität einer Kante.

**Fluss** Ein Fluss  $f$  ist eine Funktion, die jeder Kante  $e$  eines Netzwerks  $\mathcal{G}_N$  einen nichtnegativen Wert  $f(e) \leq w_e$  zuordnet und die Bedingungen (3.7) und (3.8) erfüllt. Eine Kante ist gesättigt, wenn gilt  $f(e) = w_e$ . Mit  $p \in \mathcal{V}$  (also insbesondere  $p \neq S$  und  $p \neq T$ ) gilt:

$$\sum_{(p,q) \in \mathcal{E}'} f((p,q)) = \sum_{(q,p) \in \mathcal{E}'} f((q,p)) \quad , \quad (3.7)$$

also dass Zufluss und Abfluss einander entsprechen. Es folgt

$$\sum_{(S,p) \in \mathcal{E}'} f((S,p)) = \sum_{(p,T) \in \mathcal{E}'} f((p,T)) \quad . \quad (3.8)$$

Der Betrag eines Flusses ist definiert als

$$\|f\| = \sum_{(S,p) \in \mathcal{E}'} f((S,p)) - \sum_{(p,S) \in \mathcal{E}'} f((p,S)) \quad . \quad (3.9)$$

Ein Fluss  $f^*$  ist maximal, wenn es keinen weiteren Fluss  $f'$  gibt mit  $\|f'\| > \|f^*\|$ . Ein Nullfluss ordnet jeder Kante  $e \in \mathcal{E}'$  den Fluss  $f(e) = 0$  zu.

Sogenannte *Max-Flow-Min-Cut*-Algorithmen machen sich dazu die Dualität von minimalem Schnitt und maximalem Fluss  $\|f^*\|$  zu Nutze [32], [33]:

$$\|f\| \leq \|C^*\| \quad , \quad \|f^*\| = \|C^*\| \quad (3.10)$$

Die Betrachtung von Flüssen setzt gerichtete Graphen voraus, bisher wurden jedoch nur ungerichtete Graphen betrachtet. Der Übergang ist einfach: Jede Kante in  $\mathcal{E}$  wird durch je eine Kante in Hin- und Rückrichtung ersetzt. Ein Schnitt im Sinne des vorherigen Abschnitts umfasst dann jeweils Paare gerichter Kanten, die Beziehung zwischen Formel 3.4 und dem minimalen Schnitt bleibt erhalten.

Eine wichtige Klasse von Algorithmen zur Bestimmung des maximalen Flusses, die *Augmented-Path*-Algorithmen, verwendet sogenannte zunehmende Wege  $\omega$  (*augmenting*

*Paths*) von der Quelle zur Senke eines Netzwerks. Diese sind dadurch gekennzeichnet, dass für alle Kanten  $(p, q)$  entlang des Wegs

$$f((p, q)) < w_{(p, q)} \quad (3.11)$$

gilt. Die Kanten entlang eines solchen zunehmenden Weges sind demnach nicht gesättigt, weshalb der Fluss entsprechend für alle beteiligten Kanten um

$$w^+ = \min_{(p, q) \in \omega} (w_{(p, q)} - f((p, q))) \quad (3.12)$$

erhöht werden kann. Existiert kein solcher zunehmender Weg, gibt es demnach keinen Weg von  $S$  zu  $T$ , entlang dessen der Fluss weiter erhöht werden kann. Der Betrag des gefundenen Flusses ist dann maximal. Andernfalls gilt es den bereits bekannten Fluss für alle Kanten des zunehmenden Wegs um  $w^+$  zu erhöhen und anschließend die Suche nach einem zunehmenden Weg zu wiederholen. Für endliche und diskrete Kapazitäten kann der maximale Fluss durch dieses Vorgehen in endlich vielen Iterationen ermittelt werden. Es gibt verschiedene Verfahren, die Suche nach zunehmenden Pfaden möglichst zielführend umzusetzen. Das bekannte Verfahren von Dinic findet zum Beispiel in  $\mathcal{O}(|\mathcal{V}|^2 \cdot |\mathcal{E}|)$  den maximalen Fluss eines beliebigen, gerichteten Graphen [32]. Der Spezialfall von (Pixel-) Bildern mit 4-Nachbarschaft, lässt sich im Durchschnitt effizienter lösen (siehe [32]; das beschriebene Verfahren hat allerdings eine schlechtere Worst-Case Laufzeit als der Dinic-Algorithmus).

### 3.4 $\alpha$ -Expansion Algorithmus für mehr als zwei Label

Mit  $\alpha$ -Expansion (auch Expansion-Move-Algorithmus) [29], [30] steht ein Algorithmus zur Verfügung, mit dem eine Kostenfunktion wie Formel 3.4 auch bei Vorhandensein von mehr als zwei Labeln minimiert werden kann. Das Wechselwirkungspotential muss dabei jedoch zusätzlich ein Monotoniekriterium erfüllen: Das heißt, dass für alle Label  $\alpha$ ,  $\beta$  und  $\gamma$  und für alle benachbarten Knoten  $p$  und  $q$

$$V_{p,q}(\alpha, \alpha) + V_{p,q}(\beta, \gamma) \leq V_{p,q}(\beta, \alpha) + V_{p,q}(\alpha, \gamma) \quad (3.13)$$

gelten muss. Der Ausdruck  $V_{p,q}(\alpha, \beta)$  gibt analog zu Abschnitt 3.2 das Wechselwirkungspotential zwischen den benachbarten Knoten  $p$  und  $q$  an, unter der Annahme, dass  $p$  mit  $\alpha$  und  $q$  mit  $\beta$  gekennzeichnet ist (beziehungsweise diese den entsprechenden Endknoten zugeordnet sind). Üblicherweise ist  $V_{p,q}(\alpha, \alpha) = 0$ , Formel 3.13 entspricht dann der Dreiecksungleichung (für feste  $p, q$ ).

Mit  $\alpha$ -Expansion kann nur ein Labeling zu einem lokalen Minimum der Kostenfunktion gefunden werden, das jedoch maximal um den Faktor zwei vom globalen Minimum abweicht. Der genaue Ablauf soll im folgenden nur skizziert werden. Für weitergehende Informationen sei auf [30], [31] verwiesen.

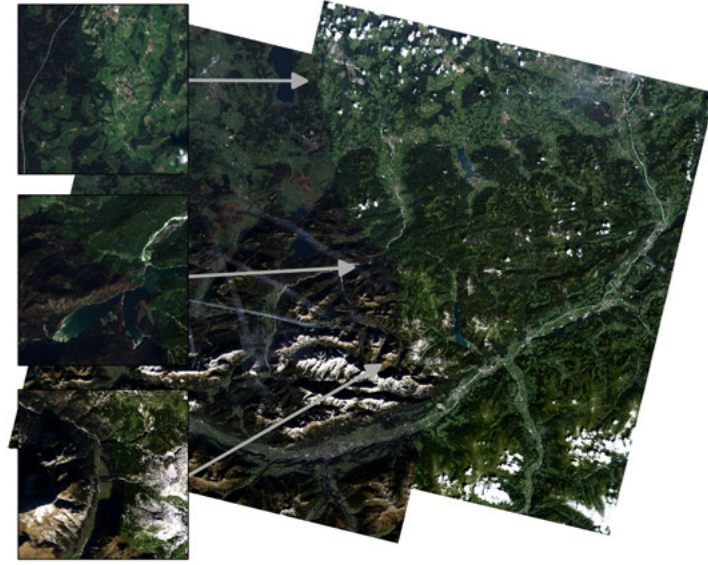
Der Algorithmus beginnt mit einem anfänglichen, beliebigen Labeling. In einer Schleife geht der Algorithmus dann wiederholt alle möglichen Label  $\alpha$  durch. In jeder Iteration wird binäres Graph-Cut-Labeling genutzt um ein neues Labeling derart zu bestimmen, dass nach der  $i$ -ten Iteration für einen Knoten  $p$  jeweils gilt,  $L_{i+1}(p) = L_i(p)$  oder  $L_{i+1}(p) = \alpha$ . Die Schleifendurchläufe werden solange wiederholt, bis in einem vollständigen Durchlauf keine Verbesserung erzielt werden konnte. Das Kriterium in Formel 3.13 stellt jeweils sicher, dass dabei keine Graphen mit negativen Kantengewichten erzeugt werden. Die Laufzeitkomplexität des Algorithmus hängt maßgeblich vom verwendeten Graph-Cut-Algorithmus für das binäre Labeling ab, die zwar allesamt polynomielle Laufzeitkomplexität aufweisen, wobei allerdings kein Verfahren mit linearer Komplexität bekannt ist.

### 3.5 Kostenfunktion

Das Optimierungsziel wird bei den präsentierten Algorithmen durch eine Kostenfunktion festgelegt, die der Form von Formel 3.4 genügen muss. Entsprechend Abschnitt 3.2, setzt sich die Kostenfunktion aus einem Gebiets- und einem Wechselwirkungspotential zusammen. Bei mehr als zwei Labeln muss außerdem für das Wechselwirkungspotential noch Formel 3.13 erfüllt sein.

Das Gebietspotential bewertet die Wahl eines Label beziehungsweise dementsprechend einer Szene, unabhängig von den Labeln der angrenzenden Pixel. Es wird nur die Umgebung des betrachteten Pixels in der zum gewählten Label gehörenden Szene betrachtet. Kriterien sind etwa die Ähnlichkeit des Tonwerts zu einem vorgegebenen Referenzwert, Kontrast- und Texturmerkmale, aber auch Metadaten wie das Aufnahmedatum des zugehörigen Bildes. In jedem Fall muss das Gebietspotential einer nichtnegativen Funktion entsprechen. Davon abgesehen, gibt es keine allgemeinen Einschränkungen.

Das Wechselwirkungspotential bestimmt Kosten für eine bestimmte Nachbarschaft. Es wird demnach bewertet, wie gut die Wahl eines Label  $\alpha$  für das Pixel  $p$  und des Label  $\beta$  für das direkt angrenzende Pixel  $q$ , „harmoniert“. Üblicherweise gilt  $V_{p,q}(\alpha, \alpha) = 0$ . Gängige Kriterien zur Beurteilung der Nachbarschaft berücksichtigen die Abweichung der Ton- (Formel 3.16) oder Gradientenwerte (Formel 3.17) der Bildpunkte  $p$  und  $q$  zwischen den Szenen  $\alpha$  und  $\beta$ . Nachfolgend sei  $\vec{I}_\alpha(p) = (I_\alpha^{(1)}, \dots, I_\alpha^{(N)})$  der Wert der



**Abbildung 3.2:** Mosaik zweier ALOS AVNIR-2 [28] Aufnahmen (2006-10-27 und 2007-07-13, radiometrisch angeglichen). Das Mosaik wurde mit der implementierten Software *FAMOS* und dem Wechselwirkungspotentials  $V_{p,q}^{(Tonwert)}$  berechnet.

Szene  $\alpha$  an der Stelle  $p$ , sowie

$$\nabla \vec{I}_\alpha(p) = \begin{pmatrix} \partial_x I_\alpha^{(1)} \\ \partial_y I_\alpha^{(1)} \\ \vdots \\ \partial_x I_\alpha^{(N)} \\ \partial_y I_\alpha^{(N)} \end{pmatrix} \quad (3.14)$$

eine geeignete Definition eines Mehrkomponenten-Gradienten. Damit lassen sich die genannten Wechselwirkungspotentiale wie folgt definieren:

$$V_{p,q}^{(Tonwert)}(\alpha, \beta) = \|\vec{I}_\alpha(p) - \vec{I}_\beta(p)\| + \|\vec{I}_\alpha(q) - \vec{I}_\beta(q)\| \quad (3.15)$$

$$V_{p,q}^{(Gradient)}(\alpha, \beta) = \|\nabla \vec{I}_\alpha(p) - \nabla \vec{I}_\beta(p)\| + \|\nabla \vec{I}_\alpha(q) - \nabla \vec{I}_\beta(q)\| \quad (3.16)$$

Die Kriterien lassen sich auch kombinieren:

$$V_{p,q}^{(T+G)} = V_{p,q}^{(Tonwert)} + V_{p,q}^{(Gradient)} \quad (3.17)$$

In Abbildung 3.2 ist ein Mosaik zweier ALOS AVNIR-2 [28] Aufnahmen dargestellt, das mit der implementierten Software *FAMOS* unter Verwendung des Wechselwirkungspotentials  $V_{p,q}^{(Tonwert)}$  berechnet wurde. Die zugrundeliegenden Aufnahmen wurden vor der

Berechnung des Mosaiks radiometrisch angeglichen (mit *Xdibias* [57]). Da die Bilder zu unterschiedlichen Jahreszeiten aufgenommen wurden (2006-10-27 und 2007-07-13) - eine gängige Ausgangssituation bei der Mosaikierung von Satellitenbildern - weisen die Aufnahmen allerdings deutliche inhaltliche Unterschiede auf. Im Ergebnis ist die gefundene Nahtlinie zwischen den Beiträgen der beiden Szenen deutlich sichtbar. In den nachfolgenden Kapiteln wird sich zeigen, dass in solchen Fällen eine Nahtlinie, die sogenannten salienten Strukturen folgt, zu bevorzugen ist (Abschnitt 4 und 5). Sei nun  $E_\alpha(p, q)$  ein beliebiges Maß dafür, wie ausgeprägt eine möglicherweise in Szene  $\alpha$  zwischen  $p$  und  $q$  verlaufende Kante ist, so ließe sich ein geeignetes Wechselwirkungspotential wie folgt definieren:

$$V_{p,q}^{(T+Salienz)}(\alpha, \beta) = \frac{V_{p,q}^{(Tonwert)}}{E_\alpha(p, q) + E_\beta(p, q)} \quad (3.18)$$

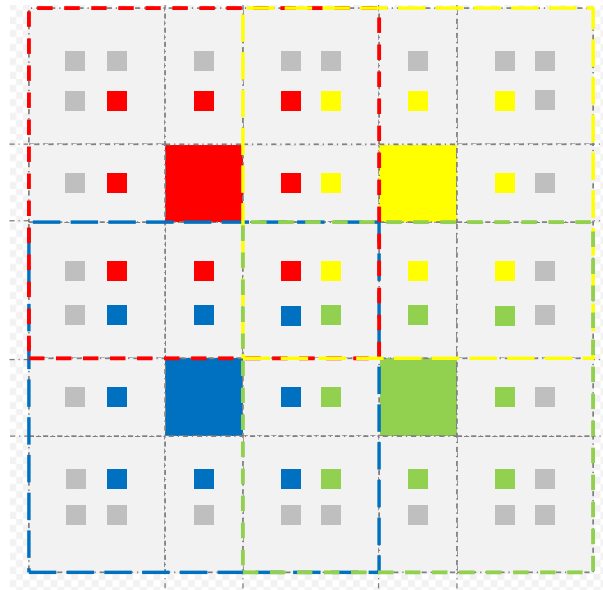
Ein solches Wechselwirkungspotential erfüllt allerdings nicht das Monotoniekriterium in Formel 3.13<sup>1</sup>. Das Beispiel zeigt somit deutlich die Grenzen der  $\alpha$ -Expansion-Methode. Das Monotoniekriterium ist dabei notwendig, um sicherzustellen, dass die Kantengewichte in den vom Algorithmus erzeugten Graphen nicht negativ werden. Es ist denkbar, solche negativen Kantengewichte zu erkennen und gegebenenfalls auf Null zu setzen. Ein solches Vorgehen wird als *Truncation* [29] bezeichnet. In jedem Fall verliert man die theoretischen Garantien bezüglich der Güte des gefundenen Labeling.

### 3.6 Iteratives Mosaikieren

*Pixel Labeling* zeichnet sich durch eine globale Herangehensweise aus. Grundsätzlich ist es demnach notwendig, bei einer Änderung des grundlegenden Datensatzes das gesamte Mosaik neu zu berechnen. Wegen der hohen Laufzeitkomplexität des  $\alpha$ -Expansion-Algorithmus gilt es jedoch eine vollständige Neuberechnung zu vermeiden. Nachfolgend wird eine Methode vorgeschlagen, die dies ermöglicht. Dazu wird das Mosaik entsprechend Abbildung 3.3 in *Zellen* unterteilt, die sich teilweise überlappen. Die Unterteilung ist sinnvollerweise so durchzuführen, dass es für jede Zelle einen *Kernbereich* gibt, der nicht von einer weiteren Zelle überdeckt wird. Geeignet sind etwa quadratische Zellen. Die Überlappungsbereiche werden dann jeweils von zwei oder vier Zellen gebildet.

Für jede Zelle kann nun ein unabhängiges Teilmosaik berechnet werden. Dabei gilt es alle Szenen zu berücksichtigen, die in den Bereich der jeweiligen Zelle hineinragen. Die Zellen werden im Sinne des  $\alpha$ -Expansion-Algorithmus durch Label repräsentiert, die unter Berücksichtigung eines zuvor festgelegten Gebiets- und Wechselwirkungspotentials (Abschnitt 3.5) den einzelnen Pixeln zugewiesen werden. Das gefundene Ergebnis ist

<sup>1</sup>Formel 3.13 läuft auf die Dreiecksungleichung hinaus (für festes  $p, q$ ), die wegen des Bruchs in Formel 3.18 jedoch nicht erfüllt wird.



**Abbildung 3.3:** Gezeigt ist eine mögliche Unterteilung des Mosaiks in quadratische Zellen, die sich teilweise überlappen. Der Kernbereich der Zellen ist farbig gefüllt dargestellt. Die verschieden gefärbten Kästchen in den Überlappungsbereichen geben an, zwischen welchen Zellen interpoliert werden muss. Graue Kästchen sind Platzhalter für mögliche weitere Zellen.

nur im Kernbereich einer Zelle für das Gesamtmosaik eindeutig. In den Ausschnitten des Mosaiks, in denen sich zwei oder vier Zellen überlappen, gilt es eine gemeinsame Lösung durch geeignetes Interpolieren zu bestimmen, wozu wiederum *Pixel Labeling* genutzt werden kann:

Die einzelnen Überlappungsbereiche bilden einen, über das gesamte Mosaik ausgedehnten, zusammenhängenden Bereich. Es ist somit nicht sinnvoll, das *Pixel Labeling* direkt auf die Vereinigung aller Überlappungsbereiche anzuwenden. Um dennoch einen kontinuierlichen Übergang zwischen den Kernbereichen zu erhalten, wird zunächst in den paarweisen Überdeckungen eine eindeutige Lösung bestimmt. Für jedes Pixel gilt es dabei aus zwei möglichen Ergebnissen zu wählen, es handelt sich demnach um ein binäres *Labeling* - je ein Label für eines der in Frage kommenden, vorläufigen Ergebnisse. Die verwendete Kostenfunktion sollte der Kostenfunktion entsprechen, die auch beim *Labeling* der Zellen genutzt wurde, allerdings wird das Gebietspotential um einen Term  $G_p^{(l)} = \text{dist}(p, \text{Kern}(l))$  ergänzt, der die Distanz des Bildpunkts zum Kernbereich des betrachteten Labels  $\text{Kern}(l)$  berücksichtigt, um einen kontinuierlichen Übergang positiv zu gewichten. Für die verbleibenden Überlappungsbereiche, an denen nunmehr je vier Zellen beteiligt sind, wird analog verfahren.

Das skizzierte Vorgehen hat zwei bedeutende Vorteile:



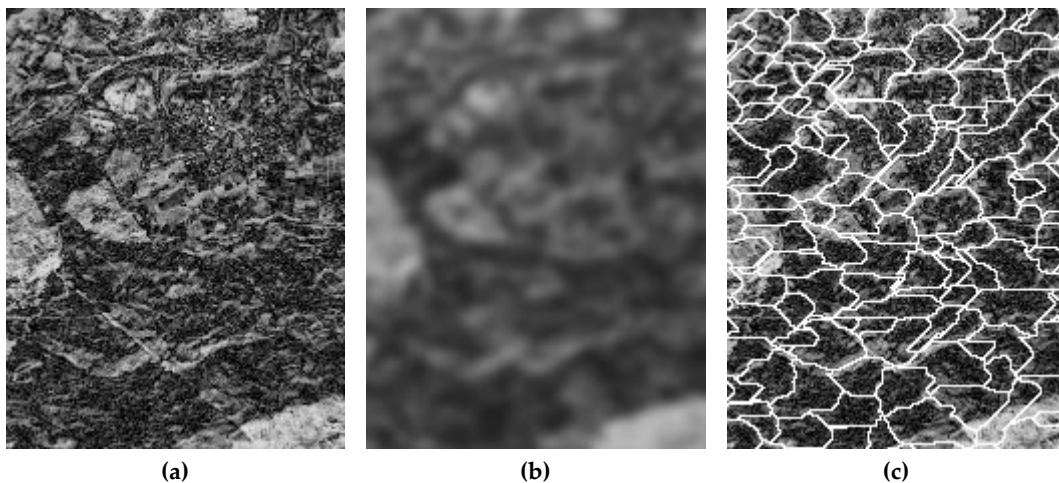
1. Ändert sich der Datensatz, aus dem das Mosaik zusammengesetzt ist, müssen nur diejenigen Zellen neu berechnet werden, in die neu hinzugekommene oder entfernte Szenen hineinragen (sowie alle Überlappungsbereiche, an denen diese Zellen beteiligt sind).
2. Die Laufzeit des  $\alpha$ -Expansion-Algorithmus ist auch von der Anzahl der zu berücksichtigenden Label abhängig. Wird das gesamte Mosaik unmittelbar berechnet, entspricht die Anzahl der Label der Anzahl der Szenen, aus denen das Mosaik zusammengesetzt werden soll. Werden die Zellen jedoch unabhängig betrachtet, muss hingegen nur ein kleiner Teil der Szenen berücksichtigt werden, deren Anzahl zwar im allgemeinen nicht nach oben beschränkt ist, in der Praxis jedoch eine gewisse Zahl nicht übersteigt.

Es ergibt sich allerdings auch ein gewichtiger Nachteil: Es ist nicht mehr möglich, die Qualität des so bestimmten Mosaiks theoretisch abzuschätzen. Die Aussage, dass die Gesamtkosten eines Mosaiks, das mit dem  $\alpha$ -Expansion-Algorithmus berechnet wurde, die Kosten eines globalen Minimums maximal um den Faktor zwei übersteigen, lässt sich nicht auf die Berechnung in unabhängigen Zellen übertragen. Bei geeigneter Wahl der Zellengrößen sowie der zugehörigen Kern- und Überlappungsbereiche, ist jedoch ein gutes Ergebnis zu erwarten.

### 3.7 Verringern der Komplexität

Angestrebt wird die Berechnung ausgedehnter Orthofotomosaiken. Wegen der sehr großen Datenmenge, sollte ein Verfahren daher eine lineare oder schlechtestenfalls log-lineare Laufzeit aufweisen. Diese Anforderung erfüllen die betrachteten Ansätze zur Umsetzung des *Pixel Labeling* nicht. Der bisher betrachtete Ansatz ordnet jeweils einzelne Pixel einer Ausgangsszene zu, was bedeutet, dass für jedes Pixel, das auf den Definitionsbereich des Mosaiks entfällt, ein Knoten in der Graph-Darstellung erzeugt wird. Tatsächlich kann es auch sinnvoll sein, jeweils Gruppen von Pixeln zusammenzufassen und durch einen einzigen Knoten zu repräsentieren, wodurch sowohl die Anzahl der Knoten als auch die der Kanten reduziert würde. Geht man dabei davon aus, dass die Anzahl der Kanten eines Pixelgraphen in der Größenordnung der Anzahl der Pixel liegt (wegen der 4-Nachbarschaft trifft dies zu) und dass diese Aussage auch im Zuge der Gruppenbildung ihre Gültigkeit behält, käme man für ein binäres *Labeling* mit dem Dinic-Algorithmus auf eine lineare Komplexität in Bezug auf die Anzahl der Pixel des Mosaiks.

Unabhängig von der gewählten Methode, die das Mosaik auf einen (vereinfachten) Graphen abbildet, bedarf es auch einer Anpassung der gewählten Kostenfunktion. Das Gebietspotential bestimmt sich (an sich unverändert) als Summe über alle Pixel einer Gruppe. Auch das Wechselwirkungspotential kann im Wesentlichen beibehalten werden,



**Abbildung 3.4:** Die effektive Komplexität des *Pixel Labeling* lässt sich verringern, in dem jeweils Gruppen von Pixeln auf einen Knoten des zugrundeliegenden Graphen abgebildet werden (siehe Text). (a) Punktwises Maximum der Tonwertunterschiede in einem Ausschnitt eines Überlappungsbereichs (invertiert) (b) Das Differenzbild aus (a) wurde mit einem Gaußfilter weichgezeichnet. (c) Ergebnis der Bildsegmentierung mit der Wasserscheidentransformation. Um in (c) erkennbare Segmente zu erhalten, wurde das Ausgangsbild in (b) übertrieben stark weichgezeichnet.

wobei nun jedoch das Wechselwirkungspotential ebenfalls über alle Pixelpaare entlang des gemeinsamen Rands der beiden betrachteten Pixelgruppen aufsummiert werden muss. Ungeachtet dessen besteht auch die Möglichkeit, ein neues Wechselwirkungspotential zu definieren, das weitergehende Kriterien wie Texturmerkmale oder andere Statistiken der betrachteten Pixelgruppen berücksichtigt.

Typischerweise überlappen sich die Szenen, aus denen das Mosaik zusammengesetzt werden soll, nicht vollständig. Es gibt demnach Bereiche des Mosaiks, für die nur je eine Szene verfügbar ist. Somit ist es naheliegend, die Größe des Graphen dadurch zu reduzieren, dass solche eindeutigen Ausschnitte des Mosaiks in einem einzigen Knoten zusammengefasst werden. Zusätzlich gilt es, geeignete Kriterien festzulegen, nach denen auch Pixel, die auf Überlappungsbereiche entfallen, gruppiert werden können. Dies muss sinnvollerweise in Abhängigkeit vom jeweiligen Optimierungskriterium, also insbesondere dem Wechselwirkungspotential geschehen.

Entspricht das Wechselwirkungspotential etwa der Tonwertdifferenz eines Szenenpaars an den zwei betrachteten Pixeln (Formel 3.16), so bietet es sich zum Beispiel an, Bereiche des Mosaiks, in denen sich die Tonwerte zwischen den verfügbaren Szenen deutlich unterscheiden, zu gruppieren. Dazu kann die Wasserscheidentransformation

(Abschnitt 5.3.1) genutzt werden<sup>1</sup>: Zunächst wird für jeden Bildpunkt des Mosaiks, also punktwise, die maximale Tonwertdifferenz aller zur Verfügung stehenden Szenen bestimmt. Dieses Differenzbild wird invertiert, gegebenenfalls weichgezeichnet und dann mit der Wasserscheidentransformation segmentiert. Das Ergebnis sind Segmente, deren Grenzen (Wasserscheiden) jeweils in Gebieten geringer Tonwertdifferenz verlaufen. Ein Beispiel ist in Abbildung 3.4 dargestellt. In Abhängigkeit vom gewählten Wechselwirkungspotential gilt es gegebenenfalls andere Kriterien als die maximale Tonwertdifferenz bei dieser Vorsegmentierung zu berücksichtigen. Das können etwa Salienzmerkmale sein, wie sie in den Abschnitten 4 und 5 geschildert werden.

---

<sup>1</sup>Diesbezüglich lohnt auch ein Blick in Abschnitt 1.3, Verwandte Arbeiten: Gracias et. al. haben ein vergleichbares Vorgehen erfolgreich zur Berechnung von Unterwassermosaiken eingesetzt [11].



# Morphological Image Compositing

---

In Kapitel 3, *Pixel Labeling*, wurde ersichtlich, dass sich das Mosaikieren als Optimierungsproblem auffassen und somit mathematisch präzise parametrisieren lässt. *Morphological Image Compositing* verfolgt hingegen einen heuristischen Ansatz, der im Gegensatz zu *Pixel Labeling* eine lineare Zeitkomplexität erwarten lässt.

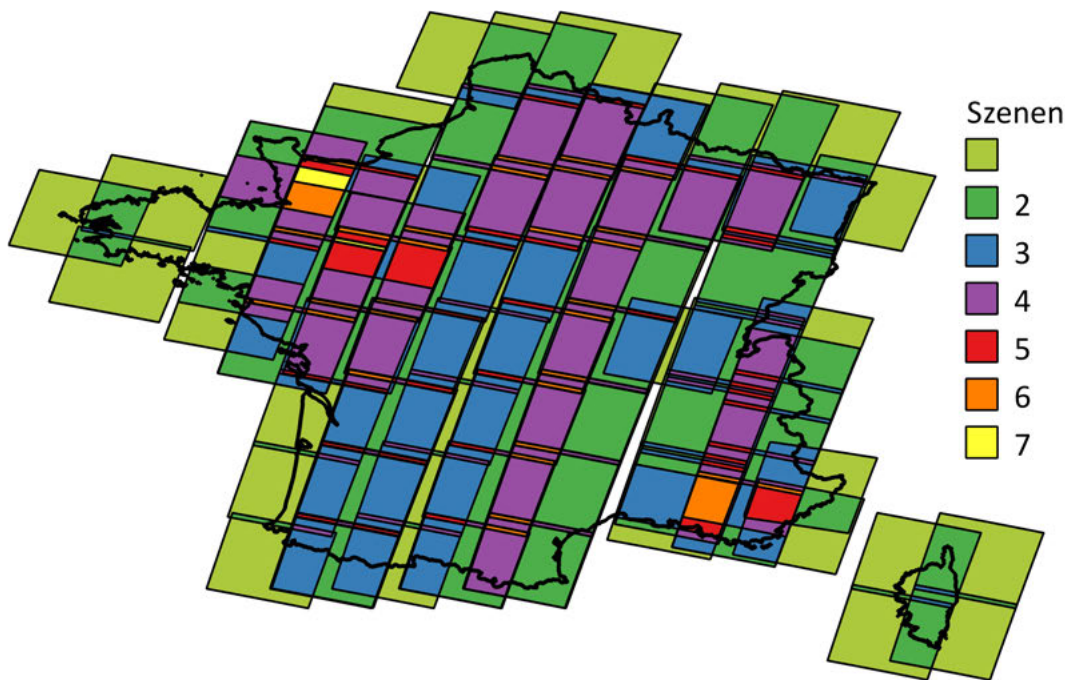
Der Algorithmus bestimmt Bereiche des Mosaiks, in denen sich die Einzelaufnahmen überlappen und unterteilt diese in disjunkte Regionen, die exklusiv einer der Aufnahmen zugeordnet werden. Die Grenzlinien zwischen den finalen Beiträgen der Szenen werden als Nahtlinie beziehungsweise Seamline bezeichnet (Formel 1.4) und sollten möglichst unscheinbar verlaufen. Bei manueller Mosaikierung von Filmabzügen in der Anfangszeit der Fernerkundung wurden gewöhnlich Einzelaufnahmen entlang von Flüssen, Straßenzügen, Schienen, Feldgrenzen oder generell salienten Geländestrukturen, die Gebiete mit unterschiedlichen Texturen beziehungsweise Tonwerten abgrenzen, beschnitten. *Morphological Image Compositing* zielt darauf ab, vergleichbare Nahtlinien in einem iterativen Prozess automatisch zu bestimmen. Ausgehend von Bereichen, in denen sich keine Szenen überlappen, werden dabei die Nahtlinien in angrenzende Regionen mit sukzessive höherem Überdeckungsgrad erweitert<sup>1</sup>. Dazu werden Bildbereiche, die zu Beginn einer Iteration bereits eindeutig einem Ursprungsbild zugeordnet werden können, in die benachbarten Überlappungsbereiche ausgedehnt (*Region Growing*). Das Verfahren ist praxiserprobt und wurde am Joint Research Center (JRC) der EU-Kommission erfolgreich zur Mosaikierung des europaweiten IMAGE-2006 Referenzdatensatzes der ESA eingesetzt [22], [23].

## 4.1 Vorüberlegungen

Dieser Abschnitt greift die in Abschnitt 1.2 eingeführte Begrifflichkeit auf und ergänzt diese. Insbesondere beziehen sich die Ausdrücke Szene und Orthofoto auf den gültigen Bildbereich und nicht das gespeicherte minimal umgebende Rechteck. Welche Anteile am finalen Mosaik aus einer bestimmten Szene stammen, steht erst fest, wenn der Algorithmus vollständig durchlaufen wurde. Mit finalem Beitrag einer Szene ist der so

---

<sup>1</sup>Dies beschreibt den Regelfall; der Sonderfall, dass einzelne Szenen vollständig überdeckt werden, ein Beitrag einer solchen Szene zum Mosaik also nicht von vornherein feststeht, wird allerdings auch berücksichtigt.



**Abbildung 4.1:** Image-2006 Abdeckung Frankreichs (Resourcesat-1 [34]) ohne Wolkenmaske: Die 109 Szenen bilden 1232 teils großflächige Überlappungsbereiche. Der Überdeckungsgrad ist farblich gekennzeichnet.

ermittelte Anteil gemeint. Gemeinsam ergeben die finalen Beiträge eine vollständige und disjunkte Aufteilung des Mosaiks.

Die nachfolgende Betrachtung geht von zwei Eigenschaften eines Datensatzes aus. Dabei sei wie in Abschnitt 1.2 der Definitionsbereich einer Szene  $\Sigma_i$ :

1. Es gibt keine zwei Szenen mit identischem Definitionsbereich:

$$\forall i \neq j: \quad \Sigma_i \neq \Sigma_j \quad (4.1)$$

2. Der Definitionsbereich einer Szene  $\Sigma_i$  soll zusammenhängend sein, das heißt für alle  $p_1, p_2 \in \Sigma_i$  gibt es einen Weg benachbarter Pixel, von  $p_1$  zu  $p_2$ . Dies vereinfacht die Betrachtung, stellt aber keine Einschränkung im eigentlichen Sinne dar: Die getrennten Bereiche können als unabhängige Szenen interpretiert werden.

In einem Mosaik werden sich im Allgemeinen Szenen überlappen, wobei Bereiche, in denen sich zwei Szenen überlappen wiederum Unterbereiche enthalten können, an denen auch eine dritte Szene beteiligt ist. Diese Betrachtung lässt sich prinzipiell bis zu beliebig hohen Überdeckungsgraden fortsetzen (Abbildung 4.1). Bereiche ohne Überlappung lassen sich als Regionen des Überdeckungsgrads eins auffassen. Die Menge der (für den

Algorithmus relevanten) Überlappungsbereiche  $\Omega$ , ergibt sich dann wie folgt. Dabei sei  $\Omega^{(l)}$  die Menge der relevanten Überdeckungen des Grads  $l$  ( $l$ -Überdeckungen) und  $\Theta(\omega)$  die Menge der zu einer Überdeckung  $\omega$  beitragenden Szenen:

$$\Theta(\omega) = \{i \mid \omega \subseteq \Sigma_i\} \quad (4.2)$$

$$\Omega^{(l)} = \{\omega = \Sigma_{i_1} \cap \Sigma_{i_2} \cap \dots \cap \Sigma_{i_l} \mid \omega \neq \emptyset \wedge \|\Theta(\omega)\| = l\} \quad (4.3)$$

$$\Omega = \bigcup_{\forall l} \Omega^{(l)} \quad (4.4)$$

Mit Voraussetzung (1) folgt direkt

$$\omega \in \Omega^{(l)}, \omega^+ \in \Omega^{(m>l)} : \quad \omega \cap \omega^+ \neq \emptyset \Rightarrow \omega^+ \subset \omega \quad , \quad (4.5)$$

also eine nach Überdeckungsgrad geschachtelte Struktur der Überlappungsbereiche. Der Bereich einer Überdeckung, der nicht in weiteren Überlappungsbereichen höheren Überdeckungsgrads enthalten ist, wird als Gültigkeitsbereich  $G(\omega)$  bezeichnet. Für eine beliebige  $l$ -Überdeckung  $\omega_l$  ist der Gültigkeitsbereich demnach

$$\Omega_l^> = \bigcup_{\forall k>l} \Omega^{(k)} \quad (4.6)$$

$$G(\omega_l) = \omega_l \setminus \bigcup_{\omega' \in \Omega_l^>} \omega' \quad . \quad (4.7)$$

$G(\omega_l) = \emptyset$  ist dabei nicht ausgeschlossen (siehe Abbildung 5.2). Darüberhinaus ist es nützlich, die Nachbarschaft  $N(\omega_l)$  einer  $l$ -Überdeckung  $\omega_l$  wie folgt zu definieren:

$$N(\omega_l) = \left\{ \omega' \in \Omega^{(k)} \mid k < l \wedge \omega_l \subset \omega' \right\} \quad (4.8)$$

## 4.2 Algorithmus

Es gilt, für Bereiche des Mosaiks, in denen sich mehrere Szenen überlappen, disjunkte Beiträge der jeweiligen Szenen zu bestimmen. *Morphological Image Compositing* macht sich dazu die geschachtelte Struktur der zuvor definierten Überlappungsbereiche  $\Omega$  zu Nutze:

1. Überlappungsbereiche  $\omega_1 \in \Omega^{(1)}$  des Überdeckungsgrads eins umfassen jeweils nur eine Szene. Eine Aufteilung in disjunkte Beiträge beteiligter Szenen entfällt somit beziehungsweise ist a priori bekannt.
2. Eine  $l$ -Überdeckung  $\omega_l$  ist echt in jedem Überlappungsbereich der Nachbarschaft  $N(\omega_l)$  enthalten (Formel 4.5). Erfolgt die Bearbeitung der Überlappungsbereiche geordnet nach dem jeweiligen Überdeckungsgrad, ist die Aufteilung aller Überdeckungen  $\omega_N \in N(\omega_l)$  in disjunkte Bereiche demnach bereits bekannt, wenn  $\omega_l$

bearbeitet wird. Aus der Nachbarschaft lassen sich daher Randbedingungen für die Aufteilung von  $\omega_l$  ableiten.

Daraus leitet sich das iterative Vorgehen von *Morphological Image Compositing* ab: Die einzelnen Überlappungsbereiche werden aufsteigend nach dem jeweiligen Überdeckungsgrad geordnet bearbeitet. Aus der Aufteilung der jeweiligen Nachbarschaft werden Keimzellen (Randbedingungen) abgeleitet, um diese anschließend mit einem Region-Growing-Verfahren auf den gesamten Überlappungsbereich auszudehnen.

#### 4.2.1 Ablauf

Der Algorithmus lässt sich in vier wesentliche Abläufe unterteilen:

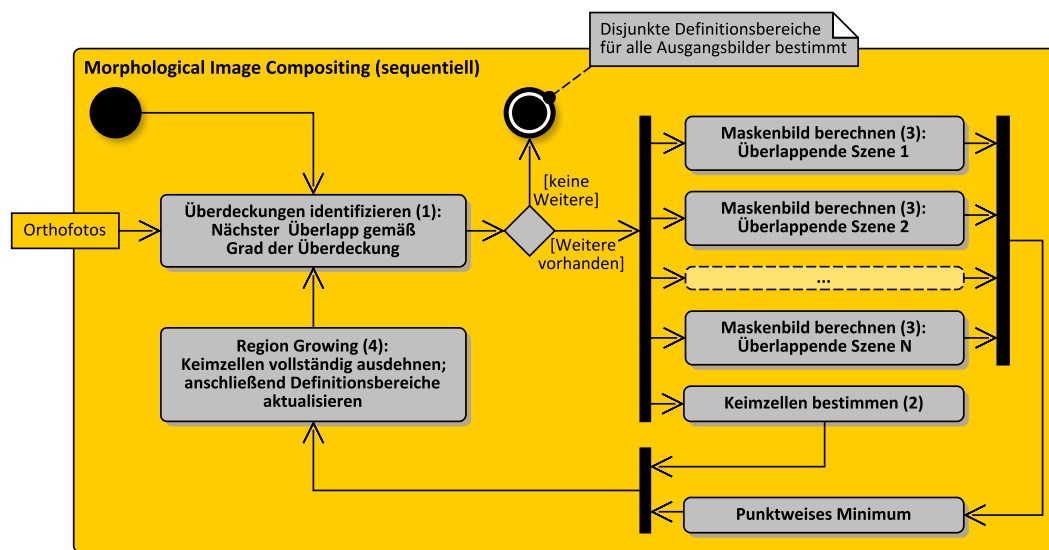
1. Identifizieren der Überdeckungen
2. Keimzellen (Randbedingungen) bestimmen
3. Maskenbild berechnen
4. Region Growing (Keimzellen ausdehnen)

*Morphological Image Compositing* führt die Unterabläufe (2) bis (4) in mehreren *Compositing*-Iterationen wiederholt aus. Nachfolgend werden die Teilprozesse des *Morphological Image Compositing* frei nach Soille [22], [23] beschrieben. Abbildung 4.2 stellt den Ablauf grafisch dar und in Abbildung 4.3 wird der Algorithmus anhand eines Beispiels illustriert.

**1) Identifizieren der Überdeckungen:** Der Algorithmus setzt voraus, dass die Überlappungsbereiche  $\Omega$  und die daran beteiligten Szenen  $\Theta(\omega \in \Omega)$  bekannt sind.  $\Theta(\omega \in \Omega)$  entspricht dabei der größtmöglichen Auswahl an Szenen, die zum jeweiligen Überlappungsbereich beitragen.

**2) Keimzellen bestimmen:** Für die Überlappungen  $\omega \in \Omega$  mehrerer Orthofotos müssen disjunkte Beiträge der beteiligten Szenen  $\Theta(\omega)$  bestimmt werden. Erfolgt die Bearbeitung der Überdeckungen aufsteigend geordnet nach dem jeweiligen Überdeckungsgrad, ist sichergestellt, dass, bezogen auf die Nachbarschaft  $N(\omega)$ , alle an die jeweils betrachtete Überdeckung angrenzenden Pixel bereits einer Szene zugeordnet wurden (Formel 4.5) und somit geeignete Randbedingungen für den Region-Growing-Prozess bekannt sind. Ausgehend von diesen Randbedingungen wird eine solche Zuordnung sukzessive auf diejenigen Pixel im Innenbereich der Überlappung ausgedehnt, deren Zuordnung noch unbekannt ist. Aus den Randbedingungen werden dazu Keimzellen abgeleitet. Dabei handelt es sich um Pixel im Randbereich der Überdeckung, deren Zuordnung zu einer Szene durch die Randbedingungen eindeutig festgelegt wird. Es gibt zwei sinnvolle Möglichkeiten diese Keimzellen zu bestimmen:





**Abbildung 4.2:** Visualisierung des *Morphological Image Compositing* als UML-Aktivitätsdiagramm. Die Ziffern in Klammern beziehen sich auf die Beschreibung des Algorithmus. Im Text werden die Schleifenabläufe Compositing-Iteration genannt.

1. Der Bereich, in dem das Region Growing angewendet wird, wird so gewählt, dass er nicht nur die betrachtete Überdeckung umfasst, in der für die Pixel noch keine Zuordnung bekannt ist, sondern ringsum jeweils noch ein angrenzendes Pixel aus bereits zugeordnetem Gebiet. Diese zusätzlichen Pixel dienen als Keimzellen.
2. Die Zuordnung zu einer Szene wird im Randbereich der Überlappung von den benachbarten Pixeln mit bekannter Zuordnung auf deren Nachbarpixel im Überdeckungsbereich übertragen, so dass diese dann als Keimzellen dienen können.

Das beschriebene Vorgehen schließt Unstetigkeiten im Randbereich der Überdeckungen aus. Dazu gehört auch, dass Orthofotos, die echt in einer weiteren Szene enthalten sind, nicht Teil des fertigen Mosaiks werden. Daraus ergibt sich auch, dass im Falle exakt deckungsgleicher Szenen keine geeigneten Keimzellen bestimmt werden können (entsprechend schließen die in Abschnitt 4.1 genannten Voraussetzungen deckungsgleiche Szenen aus). Dieser Sonderfall ist problematisch, da

gegebenenfalls „Löcher“ im Mosaik entstünden<sup>1</sup>, kann aber im Rahmen einer geeigneten (automatischen) Vorauswahl ausgeschlossen werden. Wird eine Szene vollständig von mehreren weiteren Szenen überdeckt, entspricht die Vereinigung aller paarweisen Überlappungen also dem Definitionsbereich der betrachteten Szene, lassen sich in der paarweisen Betrachtung dennoch die notwendigen Keimzellen bestimmen. Es ist somit nicht von vornherein ausgeschlossen, dass die vollständig überdeckte Szene zum finalen Mosaik beiträgt. Der Beitrag ist allerdings auch nicht zwingend erforderlich und kann demnach in Abhängigkeit vom tatsächlichen Bildinhalt entfallen (Abbildung 5.2).

**3) Maskenbild berechnen:** Das Ausdehnen der Keimzellen (Region Growing) wird durch ein Maskenbild gesteuert, das für jedes Pixel der betrachteten Überdeckung die Salienz beschreibt. Die Berechnung erfolgt zunächst unabhängig für jede zur Überdeckung beitragende Szene (*vorläufiges* Maskenbild). Um nur Strukturen zu berücksichtigen, die allen an der Überdeckung beteiligten Ausgangsbildern gemeinsam sind, wird das gemeinsame Maskenbild als punktweises Minimum der zuvor unabhängig berechneten vorläufigen Maskenbilder bestimmt. Die Salienz wird bei panchromatischen (einkanaligen) Aufnahmen mit einem morphologischen Gradienten bestimmt: Ein Strukturelement ( $SE$ ) legt die für jedes Pixel zu berücksichtigende Umgebung fest. Innerhalb dieser Umgebung wird die maximale Tonwertdifferenz bestimmt. Dazu sei  $I(p)$  der Grauwert eines einkanaligen Bilds für das Pixel  $p$ :

$$\rho_{pan}(p) = \max_{p_1 \in SE(p), p_2 \in SE(p)} (I(p_1) - I(p_2)) \quad (4.9)$$

Das Ergebnis ist eine diskrete Approximierung der Gradientenbeträge: Kanten werden hervorgehoben und homogene Flächen unterdrückt. Ein ähnliches Vorgehen ist bei mehrkanaligen Bildern möglich<sup>2</sup>. Die Pixelwerte werden dabei als Vektoren ( $\vec{I}(p)$ ) aufgefasst, für die keine definierte Ordnungsrelation existiert. Um nicht alle ( $\|SE\|_2$ ) Möglichkeiten einzeln testen zu müssen, ist die Einschränkung auf die maximale Differenz bezüglich des Zentrumspixels sinnvoll:

$$\rho_{multi}(p) = \max_{p_1 \in SE(p)} (\|\vec{I}(p_1) - \vec{I}(p)\|) \quad (4.10)$$

<sup>1</sup>Dieser Fall lässt sich einfach anhand eines Mosaiks aus nur zwei identischen Szenen verstehen: Dabei existiert genau eine 2er-Überdeckung. Der Definitionsbereich des Mosaiks beschränkt sich auf den Bereich der Überlappung. Es gibt somit keine angrenzenden Bildbereiche, aus denen Keimzellen abgeleitet werden können. Das Region Growing kann somit nicht sinnvoll durchgeführt werden; die konkrete Implementierung mittels markergesteuerter Wasserscheidentransformation liefert in diesem Fall ein ungünstiges Ergebnis, nämlich keine disjunkte Aufteilung der Überdeckung zwischen den angrenzenden Szenen, sondern leere Beiträge.

<sup>2</sup>In der Beschreibung des IMAGE-2006-Mosaiks [23] geht Soille nicht explizit darauf ein, welches Verfahren tatsächlich bei der Bestimmung des Maskenbildes verwendet wurde. Auf Seite 21 wird angedeutet, dass womöglich nur einer der verfügbaren vier Kanäle berücksichtigt wurde und die Berechnung mit dem morphologischen Gradienten erfolgte: „This gradient image is calculated by computing the point-wise minimum of the morphological gradient of the 3rd band (...)“ [23, S. 21]

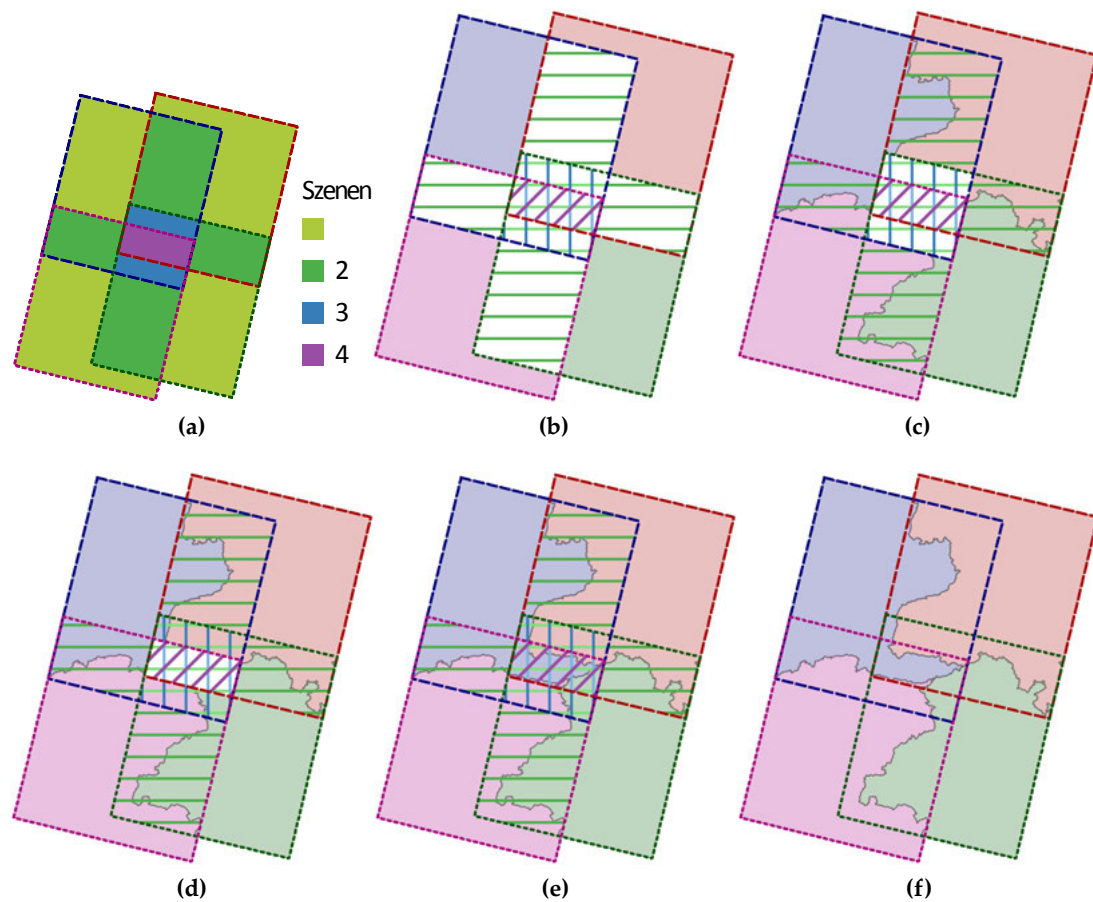
Die Größe des Strukturelements sollte jeweils entsprechend der Güte von Georegistrierung und Orthorektifizierung gewählt werden, da durch das Strukturelement festgelegt wird, wie weit die Kantenmerkmale der einzelnen Szenen jeweils auseinanderliegen können, ohne bei der Bildung des Minimums unterdrückt zu werden. Eine der Berechnung des Gradienten vorhergehende Filterung der Bilddaten kann sinnvoll sein. Soille schlägt die Verwendung eines sogenannten *Quasi-Flatzone-Filters* vor (Erläuterung in Fußnote 2 auf Seite 58) [22].

**4) Region Growing (Keimzellen ausdehnen):** Das Region Growing erfolgt auf Basis eines Verfahrens zur markergesteuerten Wasserscheidentransformation. Unter Berücksichtigung der Topografie der zu segmentierenden Bilddaten (Maskenbild) werden die Einflussbereiche festgelegter Keimzellen (Markerbild) erweitert bis das gesamte Markerbild gekennzeichnet ist. Etliche gängige Verfahren markieren darüberhinaus eine ausgezeichnete Trennlinie (Wasserscheide) zwischen den verschiedenen Einflussbereichen. Soille verwendet einen Algorithmus, der das Fluten des Maskenbildes simuliert (Vincent, Soille - „*Watersheds in Digital Spaces: an Efficient Algorithm Based on Immersion Simulations*“, 1991) [35]. Auf die grundlegende Idee hinter der Wasserscheidentransformation wird in Abschnitt 5.3.2 noch eingegangen.

Entsprechend dem Ausgang der Segmentierung werden die Beiträge der an der Überdeckung  $\omega_l$  beteiligten Szenen iterativ bestimmt, indem am Ende jeder Compositing-Iteration, bereits eindeutig der Szene zugeordnete Bereiche ergänzt werden. Eine eindeutige Zuweisung von Pixeln zu einer der beteiligten Szenen  $\Theta(\omega_l)$  ist dabei nur innerhalb des jeweiligen Gültigkeitsbereichs  $G(\omega_l)$  möglich, also nur für Pixel, die nicht auch in einer Überlappung  $\omega^+ \in \Omega^{(>l)}$  höheren Grads enthalten sind. Für diese Regionen kann ein eindeutiges Ergebnis erst in einer späteren Compositing-Iteration bestimmt werden: Da es sich bei der Wasserscheidentransformation jedoch um ein globales Verfahren handelt, ist die Einbeziehung dieser Bereiche in das Region Growing dennoch unumgänglich - ein Teil des Ergebnisses muss gegebenenfalls verworfen werden.

Keimzellen können wie beschrieben bei aufsteigend nach Überdeckungsgrad geordneter Betrachtung der Überlappbereiche zuverlässig bestimmt werden: Im Ausgangszustand sind nur die Pixel eindeutig bestimmt, die in keine der identifizierten Überdeckungen fallen. Durch Ausdehnen der daraus abgeleiteten Keimzellen in die Überdeckungen des Grades zwei können anschließend Pixel, die maximal in Überlappungsbereichen des Grades zwei enthalten sind, eindeutig einer Szene zugeordnet werden. Das Verfahren wird entsprechend für alle höheren Überdeckungsgrade fortgesetzt, bis keine unbearbeiteten Überdeckungen verbleiben.

Zur Identifizierung der Überdeckungen verwendet Soille eine vorberechnete (symmetrische) Überlappmatrix (*overlap matrix*), die für jedes Szenenpaar das Vorhandensein eines paarweisen Überlapps angibt. Bei sequentieller Verarbeitung ergibt sich die Reihenfolge, in der die Überdeckungen bearbeitet werden, aus der zeilenweisen Betrachtung



**Abbildung 4.3:** (a) Das Mosaik besteht aus vier Szenen. Die Überdeckungsgrade sind farblich gekennzeichnet: Es treten fünf 2er- (die fünfte Überdeckung ist [ ]), zwei 3er- und eine 4er-Überdeckung auf. Alle weiteren Überlappungen (etwa [ ] und [ ]) sind kongruent zu einer der 3er- oder 4er-Überdeckung. (b) Die Beiträge der vier Szenen sind für die Bereiche mit Überdeckungsgrad eins bekannt und legen die Keimzellen für die Unterteilung der fünf Überlappungen des Grads zwei in disjunkte Beiträge fest. (c) Die bekannten Keimzellen wurden in die Überdeckungen des Grads zwei propagiert (nicht dargestellt für [ ]). Da die Wasserscheidentransformation global wirkt, betrifft die Segmentierung jeweils auch den Bereich der eingeschlossenen Überdeckungen höheren Grades (noch weiß). Dort ist das Ergebnis nicht eindeutig, allerdings ist nun die direkte Umgebung der beiden 3er-Überlappungen ( -schraffiert) bekannt und es können im nächsten Schritt entsprechende Keimzellen abgeleitet werden. In zwei weiteren Compositing-Iterationen wurden disjunkte Beiträge für die verbleibenden Überlappungen der Grade drei (d) und vier (e) bestimmt. (f) Das finale Mosaik.

der Überlappmatrix. Im Zuge dessen werden auch ausstehende Überdeckungen höheren Grades identifiziert, die in weiteren analogen Durchläufen betrachtet werden. Darüberhinaus erörtert Soille ein einfaches Schema zur parallelen Berechnung des Mosaiks. Dabei werden unabhängige Szenen separat und parallel bearbeitet. Im folgenden sei

$$\Pi(\sigma) = \bigcup_{\forall \Sigma \mid \Sigma \cap \sigma \neq \emptyset} \Sigma \quad (4.11)$$

der Definitionsbereich einer Szene, vereinigt mit den Definitionsbereichen aller überlappenden Orthofotos. Zwei Szenen  $\Sigma_i$  und  $\Sigma_j$  werden nun als unabhängig betrachtet, wenn  $\Pi(\Sigma_i) \cap \Pi(\Sigma_j) = \emptyset$ . Diejenige Menge unabhängiger Szenen mit maximaler Kardinalität zu finden ist nicht ohne weiteres möglich, weshalb lediglich *eine* Auswahl an Szenen mit dieser Eigenschaft bestimmt wird. Da es vorkommen kann, dass in einer solchen Auswahl nicht alle zu mosaikierenden Szenen enthalten sind, muss der Vorgang nach der Prozessierung der ausgewählten Szenen wiederholt werden, bis keine weiteren Orthofotos verbleiben. Jede der unabhängigen Szenen wird analog zum sequentiellen Algorithmus verarbeitet.

### 4.3 Vor- und Nachteile

Bei näherer Betrachtung des *Morphological Image Compositing* lassen sich einige Vor- und Nachteile identifizieren, die in dieser Form von Soille jedoch nicht näher betrachtet werden.

**Pro: Unkompliziert** *Morphological Image Compositing* besteht aus leicht nachvollziehbaren Abläufen. Der Algorithmus lässt sich ausschließlich mit Standardoperationen auf Rasterbildern implementieren. Das gilt auch für die Bestimmung der Keimzellen, vorausgesetzt die Pixelmasken der Definitionsbereiche werden in Maskenbildern verwaltet. Die Beschreibung in [22] legt nahe, dass dieser Ansatz gewählt wurde.

**Pro: Iteratives Mosaikieren prinzipiell möglich** Werden Szenen zu einem bereits mosaikierten Datensatz hinzugefügt oder aus diesem entfernt, können in beiden Fällen Überdeckungen hinzukommen oder entfallen. Neben dem Identifizieren dieser Änderungen ist im Prinzip nur die Segmentierung neu entstandener Überdeckungen erforderlich, um das Mosaik zu aktualisieren.

**Pro: Erweiterbar** Durch Anpassungen bei der Bestimmung des Maskenbilds können auch andere Kantenmerkmale sowie weitere Kriterien berücksichtigt werden.

**Kontra: Ignoriert paarweise Nahtlinien** Die zu segmentierenden Maskenbilder werden als punktweises Minimum der für jedes Ausgangsbild individuell berechneten, *vorläufigen* Maskenbilder gebildet. Insofern berücksichtigt *Morphological Image*

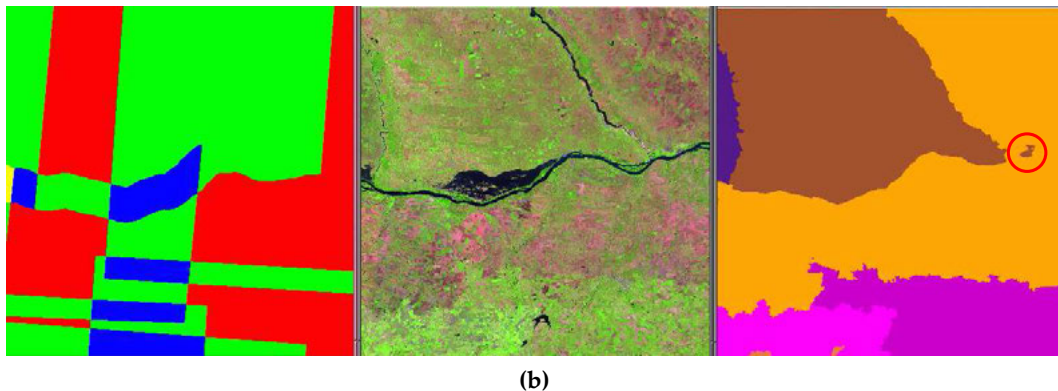
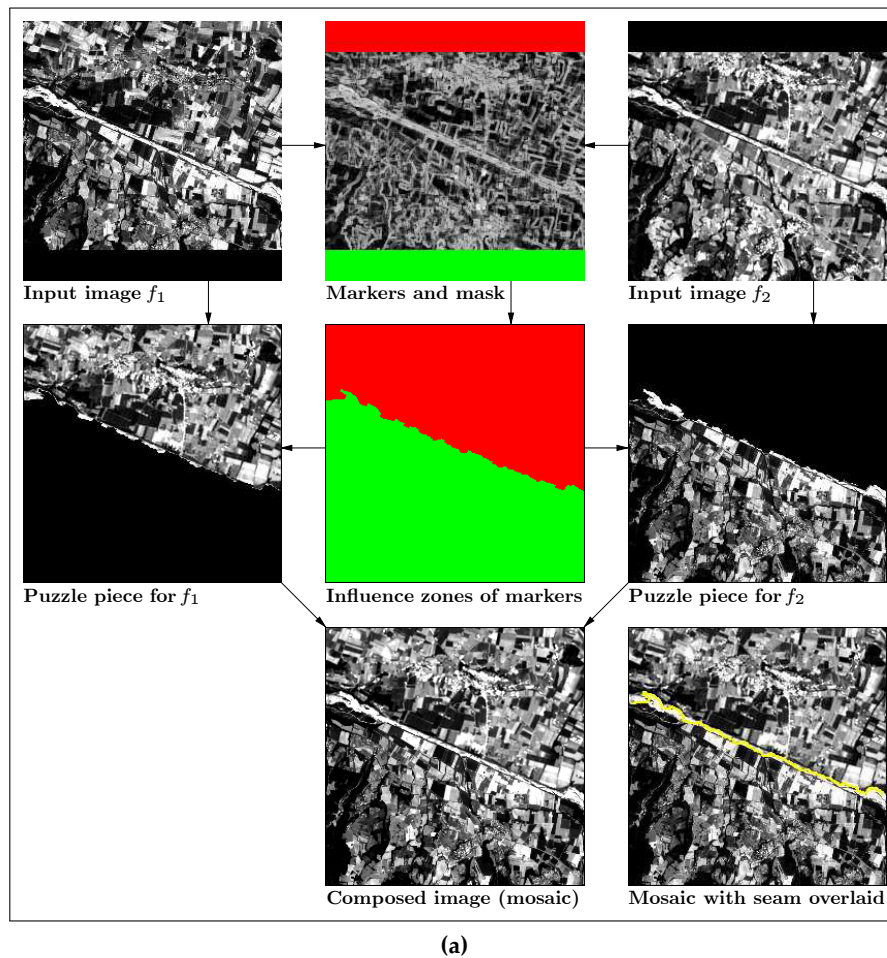
*Compositing* nur jene Kanten, die in allen beteiligten Szenen auftreten. Tatsächlich grenzen die gesuchten Nahtlinien aber nur je zwei Szenen voneinander ab (abgesehen von den Vereinigungspunkten mehrerer Seamlines), weshalb die Betrachtung paarweise auftretender Kanten besser geeignet wäre, um die Nahtlinien zuverlässig entlang salienter Strukturen zu platzieren. Die ausschließliche Betrachtung des Minimums trägt jedoch maßgeblich zum unkomplizierten Aufbau des Algorithmus bei. Erfolgt die Aufnahme der Orthofotos in teilweise überlappenden Streifen, relativiert sich dieser Nachteil jedoch: Dann überwiegt der Flächenanteil der Zer-Überdeckungen.

**Kontra: Einfaches Kantenmerkmal** *Morphological Image Compositing* verwendet einen *morphologischen Gradienten* zur Approximation der Gradientenmagnitude. Dieses Maß für die Stärke der lokalen Bildänderung wird als ausschließliches Kantenmerkmal verwendet. Weitere Kriterien, wie die Orientierung oder Länge möglicher Kanten, bleiben unberücksichtigt.

**Kontra: Mäßig präzise Nahtlinien** Aufgrund des einfachen Kantenmerkmals und der ausschließlichen Berücksichtigung des punktwisen Minimums der unabhängig bestimmten, vorläufigen Maskenbilder, ergibt sich ein teilweise nur mäßig präziser Verlauf der Nahtlinien. In Abbildung 4.4 ist ein Beispiel aus der Veröffentlichung von Soille [23] dargestellt: Die Seamline folgt zwar wie gewünscht dem markanten Flußlauf, verläuft dabei aber nicht präzise entlang einer der Uferlinien.

**Kontra: Artefakte** Ein mit *Morphological Image Compositing* erzeugtes Mosaik kann „isolierte Bereiche“ enthalten: Dabei besteht der Beitrag einer Szene aus mehreren unzusammenhängenden Regionen. Dies kann bei Vorhandensein als ungültig markierter Bildbereiche (etwa Wolken) erwünscht und notwendig sein, im Allgemeinen gilt es jedoch unnötige Nahtlinien zu vermeiden. Auf die Entstehung der Artefakte wird in Abschnitt 5 näher eingegangen. Siehe auch Abbildung 4.4b (Artefakt mit rotem Kreis markiert).

**Kontra: Ergebnisse werden verworfen** Die Wasserscheidentransformation ist ein globales Verfahren und muss auf den gesamten Überdeckungsbereich angewandt werden, aber nur ein Teil des Ergebnisses kann berücksichtigt werden.



**Abbildung 4.4:** (a) Mäandernde Nahtlinie. (b) Isolierter Bereich (rot markiert): Die verschiedenen Farben repräsentieren Beiträge unterschiedlicher Szenen. Im linken Bild ist der Überdeckungsgrad dargestellt (■ =1, ■ =2, ■ =3). Die Abbildungen dienen der Diskussion des *Morphological Image Compositing* anhand der Ergebnisse von Soille (siehe Text) und wurden dazu einem Report des Joint Research Center (JRC) der europäischen Kommission entnommen („*The IMAGE-2006 Mosaic Project*“, 2011 - Seite 86 und Seite 105) [23]. Der rote Kreis in (b) wurde ergänzt. Die Nutzung erfolgt im Rahmen des Kommissionsbeschlusses über die Weiterverwendung von Kommissionsdokumenten [36].





### Implementiertes Verfahren: MSeam

---

Viele verwandte Arbeiten beschäftigen sich mit fundierten Methoden zur Bestimmung paarweiser Nahtlinien. In der Praxis besteht ein Orthofoto-Mosaik allerdings aus einer größeren Anzahl Szenen, wobei sich lokal mehr als zwei Fotos überdecken. Zusätzlich ist das bei Fernerkundungsdaten anfallende, sehr große, Datenvolumen zu berücksichtigen, wodurch einige elaborierte Verfahren nur eingeschränkt anwendbar sind (zum Beispiel *Pixel Labeling*, siehe Abschnitt 3). Auch kann es von Interesse sein, ein bestehendes Mosaik zu aktualisieren, wobei sich das Ergebnis nicht von dem Mosaik unterscheiden sollte, das bei sofortiger Mosaikierung des finalen Datensatzes entstanden wäre. Nachfolgend wird in diesem Zusammenhang von iterativem Mosaikieren gesprochen - nicht zu Verwechseln mit der im vorigen Abschnitt beschriebenen wiederholten Ausführung von *Compositing*-Iterationen im Rahmen des *Morphological Image Compositing*. Beim Entwurf von *MSeam* wurde das iterative Mosaikieren von vornherein berücksichtigt, wohingegen keine der in Abschnitt 1.3 genannten, verwandten Arbeiten auf diesen Aspekt eingeht.

*MSeam* wurde maßgeblich von *Morphological Image Compositing* (Soille) [22], [23] inspiriert, das im vorigen Abschnitt besprochen wurde. Die Möglichkeit des iterativen Mosaikierens wurde von Soille zwar nicht berücksichtigt, bei näherer Betrachtung stellt sich jedoch heraus, dass sich die zugrundeliegende Methodik auch für ein iteratives Vorgehen eignet. Dazu ist es sinnvoll, die Beiträge der einzelnen Szenen erst in einem abschließenden *Renderschritt* zu bestimmen, was auch das Filtern möglicher Artefakte erleichtert. Somit wird allerdings die effiziente Speicherung von Zwischenergebnissen, hier, in Form von Vektordaten notwendig. Das ursprüngliche Verfahren arbeitet nahezu ausschließlich mit Rasterdaten, auch die Definitionsbereiche der einzelnen Orthofotos werden als Markerbild (*Labelmatrix*) repräsentiert. Neben der effizienten Speicherung von Pixelmasken ermöglicht die Nutzung von Vektordaten auch die Anwendung von Methoden der algorithmischen Geometrie anstelle der ausschließlichen Verwendung von Pixeloperationen. Es gibt etliche ausgereifte Datenbanksysteme mit Unterstützung für Geodaten, die zur Verwaltung der anfallenden Daten genutzt werden können. Ein Mosaik wird dann durch den jeweiligen Datenbankzustand bestimmt. Je nach gewünschter Nutzung des Mosaiks kann dieses Ergebnis bereits ausreichend sein oder wird zur Ausgabe eines oder mehrerer Rasterbilder genutzt.

Im Rahmen der Masterarbeit ist die Software *FAMOS - Fully Automated Mosaicking Study* entstanden, auf der *MSeam* aufbaut. Diese besteht aus einem erweiterbaren Framework,

das neben einer Implementierung grundlegender Vektoroperationen auch unkompliziertes *In-Memory* und *On-Disk* Caching, einige spezielle Bildverarbeitungsroutinen und weitere nützliche Funktionen bereitstellt. Darüberhinaus enthält das Framework allgemeine und erweiterbare Komponenten zur Verwaltung des Mosaikierungszustands, die auf der *In-Process*-Datenbank *SQLite* [58] mit *SPATIALITE*-Erweiterung [52] aufbauen. *FAMOS* stellt die grundlegenden Funktionalitäten wie die Indexierung georeferenzierter *Xdibias*-Orthofotos [57] und das Identifizieren von Überdeckungen zur Verfügung. Konkrete Verfahren zur Mosaikierung werden als *MosaickingStrategy* bezeichnet und können flexibel ergänzt werden. Zur Nutzung solcher Strategien stellt *FAMOS* eine *Python*-API [516] und ein Kommandozeilenschnittstelle zur Verfügung. Letzteres verwendet eine *XML*-Steuerdatei zur Festlegung der Prozessierungsparameter und der zu verwendenden Strategie. Neben *Pixel Labeling* und dem *einfachen Übereinanderlegen* steht schwerpunktmäßig eine, im Rahmen dieser Arbeit entwickelte Strategie für *MSeam* zur Verfügung.

Dieses Kapitel baut auf der Schilderung des *Morphological Image Compositing* in Abschnitt 4 auf. Die dort gebrauchte Begrifflichkeit wird fortgeführt.

## 5.1 Von *Morphological Image Compositing* zu *MSeam*

*Morphological Image Compositing* macht sich zu Nutze, dass in Bereichen, in denen sich keine Szenen überlappen, eine eindeutige Zuordnung zu den jeweiligen Ursprungsbildern vorgegeben ist (bezogen auf die jeweilige Nachbarschaft, siehe Kapitel 4). Diese Zuordnung wird in mehreren Compositing-Iterationen sukzessive auf angrenzende Regionen höheren Überdeckungsgrads ausgedehnt. Werden die einzelnen Überdeckungen geordnet nach dem jeweiligen Überdeckungsgrad betrachtet, ist jeweils sichergestellt, dass angrenzende Bildbereiche bereits eindeutig einer Szene zugeordnet wurden. Die nach Überdeckungsgrad geschachtelte Struktur der Überlappungsbereiche legt die Reihenfolge der Compositing-Iterationen nur teilweise fest, weshalb diese im Rahmen der vorhandenen Abhängigkeiten frei bleibt. Im Zuge des *Scheduling* können dementsprechend weitere Kriterien berücksichtigt werden, wie etwa der Zustand des *Caches* oder die Verfügbarkeit lokal zwischengespeicherter Orthofotos.

In jeder Compositing-Iteration werden Keimzellen bestimmt, die auf den Bereich der jeweils betrachteten Überdeckung ausgedehnt werden. Dieses *Region Growing* wird als markergesteuerte Wasserscheidentransformation umgesetzt. Da es sich dabei um ein global arbeitendes Verfahren handelt, wird im Zuge dessen, die jeweilige Überdeckung vollständig unter den beteiligten Szenen aufgeteilt, obwohl das Ergebnis wegen der geschachtelten Struktur der Überlappungsbereiche nur für einen Teil der Überdeckung eindeutig ist (Gültigkeitsbereich, siehe Formel 4.7). Bei *Morphological Image Compositing* werden die sich zunächst ergebenden Anteile der Aufteilung, die außerhalb des Gültigkeitsbereichs liegen, im weiteren Verlauf verworfen.

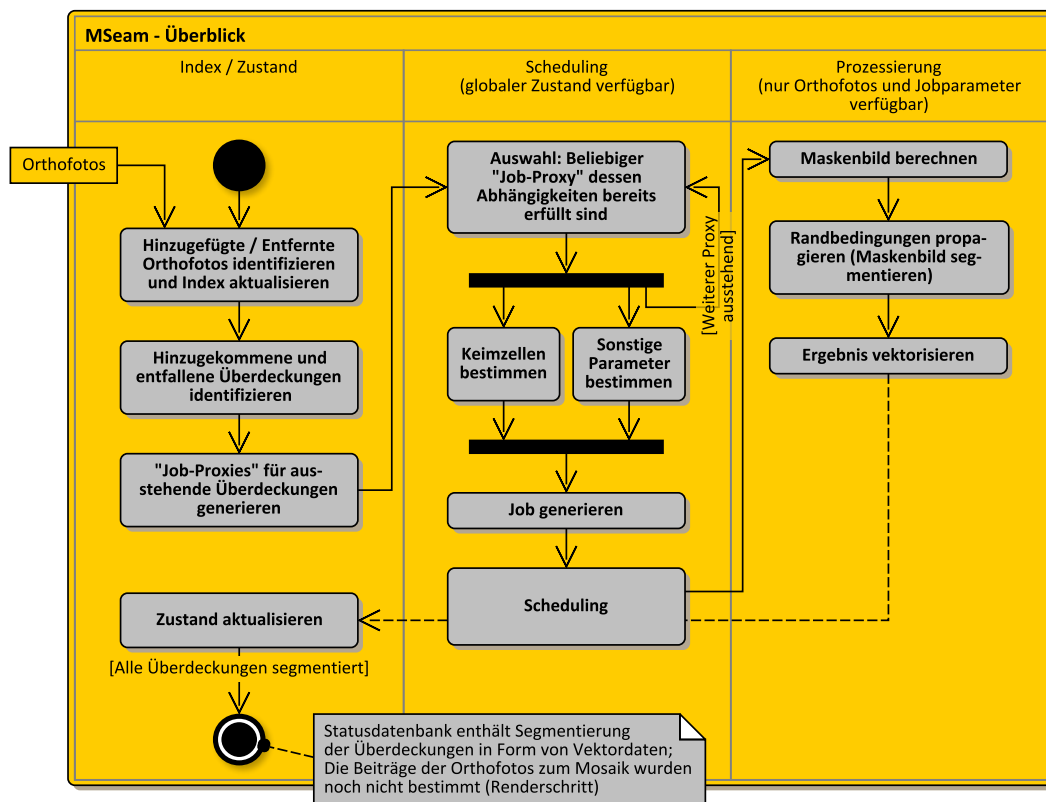
*MSeam* unterscheidet sich von *Morphological Image Compositing* durch die weitreichende Verwendung von Vektordaten beziehungsweise algorithmischer Geometrie an Stelle von Pixeloperationen. Der grundlegende Ablauf der Compositing-Iterationen wurde beibehalten, aber in Teilen unter Berücksichtigung einer möglicherweise verteilten Berechnung ergänzt (Abbildung 5.1): Insbesondere werden zunächst nur die Überdeckungen in geeigneter Reihenfolge bearbeitet, wobei die Ergebnisse in Form von Vektordaten vollständig gespeichert werden. Die Bestimmung der Beiträge einzelner Szenen verlagert *MSeam* in einen abschließenden Renderschritt, wohingegen *Morphological Image Compositing* die Beiträge als Markerbild verwaltet und diese im Zuge jeder Compositing-Iteration aktualisiert. Das eigentliche Region Growing wird identisch gehandhabt. Dementsprechend wird der Verlauf der Nahtlinien maßgeblich durch das Maskenbild bestimmt, weshalb dessen Berechnung in einen erweiter- und steuerbaren Teilprozess ausgliedert wurde.

Im Gegensatz zu *Morphological Image Compositing* verwendet *MSeam* einen Index der zu mosaikierenden Szenen. Unter Verwendung eines räumlichen R-Baum-Index in Kombination mit einem *Sweep*-Verfahren ermöglicht dies das effiziente Identifizieren der relevanten Überdeckungen. Einzelne Compositing-Iterationen werden zu unabhängigen Arbeitspaketen („*JobProxy*“) mit Beschreibung aller vorhandenen Abhängigkeiten zusammengefasst, was eine flexible Anpassung der Reihenfolge der einzelnen Compositing-Iterationen gestattet. Steht ein Arbeitspaket zur Bearbeitung an, wird daraus ein serialisierbarer Arbeitsauftrag („*Job*“) erzeugt. Ein solcher „*Job*“ enthält alle notwendigen Informationen, die für die Bestimmung des Maskenbilds und die Durchführung des Region Growing erforderlich sind, insbesondere eine Beschreibung der Keimzellen. Somit ist es möglich, den Arbeitsauftrag ohne Zugriff auf die Statusdatenbank zu bearbeiten. Dies ist zum Beispiel dann interessant, wenn der Mosaikierungszustand in einer *In-Process*-Datenbank verwaltet wird, auf die die Rechenknoten keinen Zugriff haben.

Abbildung 5.1 vermittelt einen Überblick der Funktionsweise von *MSeam*. Für eine etwas detailliertere Betrachtung der Implementierung, insbesondere auch in Hinblick auf die vektorielle Abstraktion von Pixelmasken, sei an dieser Stelle auf Anhang A verwiesen.

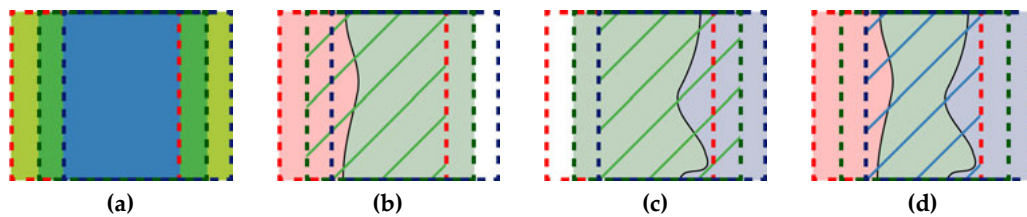
## 5.2 Iteratives Mosaikieren

Wird der Umfang des zu mosaikierenden Datensatzes durch das Hinzufügen oder Entfernen von Szenen verändert, können jeweils neue Überdeckungen entstehen und bereits segmentierte Überlappungsbereiche entfallen oder ersetzt werden. Beim Identifizieren entsprechender Änderungen ist nicht nur die Ausdehnung des Überlapps zu berücksichtigen, sondern auch die Menge der beteiligten Szenen (Abbildung 5.3). Alle neu hinzugekommenen Überdeckungen werden analog zu den üblichen Compositing-Ita-

Abbildung 5.1: Funktionsweise von *MSeam*

tionen unter Berücksichtigung der Abhängigkeiten segmentiert, wobei wiederum nur ein Teil des Ergebnisses eindeutig ist. Da die Ergebnisse der Compositing-Iterationen vektoriell gespeichert werden, ist es jedoch problemlos möglich, das vollständige Resultat zu speichern. Dies kann sich insbesondere beim Entfernen von Szenen als zweckmäßig erweisen, da dann potentiell einige Compositing-Iterationen eingespart werden (Abbildung 5.3). Ist die Aufteilung aller Überdeckungen in disjunkte Beiträge bekannt, lässt sich daraus direkt der Beitrag jeder Szene zum Mosaik ableiten. Dank der vektorbasierten Speicherung dieser Informationen ist die Berechnung unproblematisch und erfolgt während der Ausgabe des Mosaiks. Die Speicherung redundanter Daten wird somit vermieden.

Wird der zugrundeliegende Datensatz fortwährend um zusätzliche Aufnahmen des bereits mosaikierten Gebiets erweitert, hat dies zwangsläufig eine Mehrfachabdeckung zur Folge. Aufgrund der in Abschnitt 4.2 beschriebenen Einschränkungen, muss bei Vorhandensein von Orthofotos mit identischem Definitionsbereich (Wolkenmaske bereits berücksichtigt) jeweils eine representative Szene ausgewählt werden. Unproblematisch sind hingegen Szenen, die vollständig überdeckt werden, also vollständig in der Vereinigung aller Überdeckungen enthalten sind. Ein Beitrag zum fertigen Mosaik

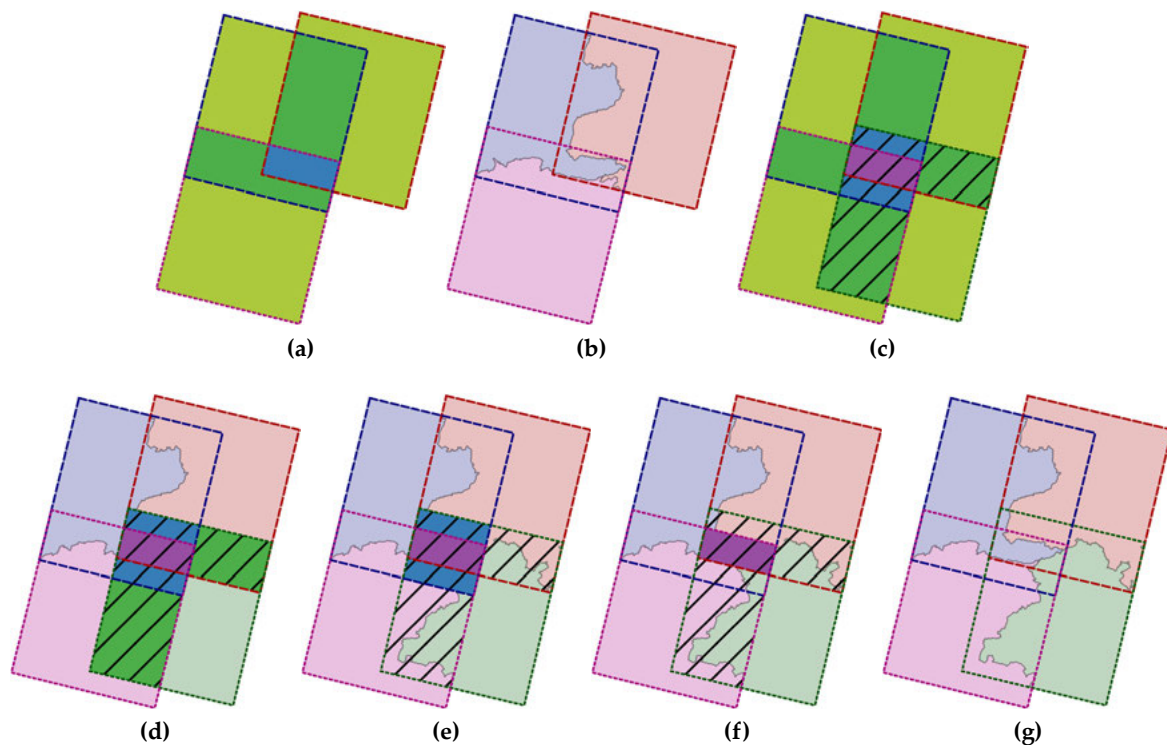


**Abbildung 5.2:** Dieses Beispiel illustriert, dass ein Beitrag von Szenen, die vollständig überdeckt werden, nicht notwendigerweise gewährleistet ist: Die Ausgangssituation (a) besteht aus drei Szenen, wobei die mittlere Szene (■) vollständig von den äußeren Szenen überdeckt wird. Die Überdeckungsgrade sind farblich gekennzeichnet (■ = 2, ■ = 3). In (c) und (d) ist eine mögliche Unterteilung der beiden 2er-Überdeckungen dargestellt. In diesem Beispiel wurde der jeweilige Gültigkeitsbereich (Bereich, der nicht auch in die 3er-Überdeckung fällt) vollständig der linken beziehungsweise rechten Szene zugeordnet. In (d) ist das resultierende Zwischenergebnis unmittelbar vor der Aufteilung der verbleibenden 3er-Überdeckung (schraffiert) dargestellt. Im Zuge des Region Growing wird die 3er-Überdeckung demnach ausschließlich unter der linken und rechten Szene aufgeteilt werden, da keine Keimzellen beziehungsweise Randbedingungen einen Anteil der mittleren Szene notwendig machen. Dementsprechend wird die mittlere Szene keinen Beitrag zum fertigen Mosaik leisten.

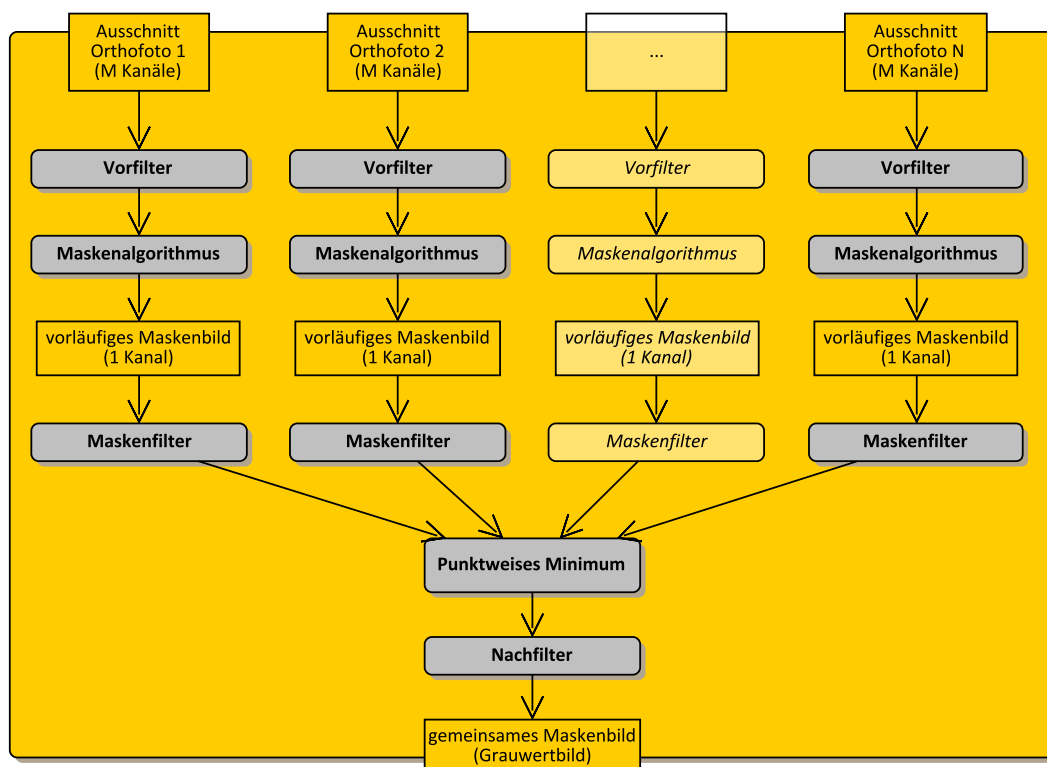
wird in diesem Fall jedoch nicht gewährleistet und ist abhängig von Bildmerkmalen (Abbildung 5.2). Tatsächlich kann es sogar erwünscht sein, nicht jede Szene ins Mosaik aufzunehmen um viele kleine Beiträge zu vermeiden. Dies kann gegebenenfalls in Form einer externen Vorauswahl umgesetzt werden.

### 5.3 Das Maskenbild

Ein sogenanntes Maskenbild (Abbildung 5.8) steuert das Region Growing und bestimmt damit maßgeblich die Aufteilung der Überdeckungen in disjunkte Beiträge beteiligter Szenen, also letztlich die Platzierung der Nahtlinien. Angestrebt wird demnach eine Kombination aus Region-Growing-Verfahren und Maskenbild, die Seamlines an salienten Bildmerkmalen ausrichtet. Dabei gilt es, geometrische, radiometrische und inhaltliche Unterschiede der sich überdeckenden Szenen angemessen zu tolerieren. Im Gegensatz zu *Morphological Image Compositing* wird das Maskenbild von *MSeam* in einem mehrstufig gegliederten Prozess berechnet (Abbildung 5.4), der vielfältig beeinflusst werden kann. Dazu kann jeder Teilschritt durch Zusammenstellen vordefinierter Filter und Operationen individuell angepasst werden. Zusätzlich ermöglicht eine *Python-API* die Ergänzung weiterer Filter und Operationen. Davon abgesehen definiert *MSeam* al-



**Abbildung 5.3:** (a) Das anfängliche Mosaik besteht aus drei Szenen mit der in (b) gezeigten disjunkten Aufteilung. Die Überdeckungsgrade sind farblich gekennzeichnet (■=2, ■=3, ■=4). In (c) wurde das Mosaik um eine vierte Szene erweitert: Im schraffierten Bereich sind somit neue Überdeckungen entstanden ([■ ■], [■ ■], [■ ■], [■ ■], [■ ■], [■ ■]). Die Überdeckung des Grads vier ersetzt den bisherigen 3er-Überlappbereich ([■ ■]), dessen Segmentierungsergebnisse somit verworfen werden; die Unterteilung der verbleibenden 2er-Überdeckungen kann beibehalten werden. Die neue Ausgangssituation ist in (d) abgebildet. Die bekannten Keimzellen werden analog zu Abbildung 4.3 propagiert (e-g). Die finalen Beiträge der Szenen zum Mosaik sind in (g) dargestellt. Soll umgekehrt die soeben hinzugefügte Szene wieder entfernt werden, entfallen alle Überdeckungen, die vollständig im schraffierten Bereich liegen (siehe oben) und es kommt der in (a) gezeigte 3er-Überlappungsbereich hinzu. Auch in diesem Beispiel wird somit eine Compositing-Iteration erforderlich. Es erweist sich dabei als zweckmäßig die Ergebnisse der Unterteilung einer Überdeckung vollständig gespeichert zu haben (nicht nur den eindeutig bestimmten Bereich): Ansonsten hätten im Anschluss an die Entfernung der grünen Szene auch die beiden verbleibenden 2er-Überdeckungen neu segmentiert werden müssen, da der eindeutig bestimmte Bereich in (a) im Vergleich zu (c) größer ausfällt.



**Abbildung 5.4:** Das Region Growing, also das Ausdehnen der Beiträge einzelner Szenen in den Compositing-Iterationen, wird durch ein Maskenbild gesteuert. Dabei handelt es sich um ein Grauwertbild, dessen Intensitätswerte die Salienz des jeweiligen Bildpunkts angeben. Das einkanalige Maskenbild wird in einem mehrstufigen Ablauf aus  $M$  mehrkanaligen Ausschnitten der zur Überdeckung beitragenden Orthofotos berechnet (Abschnitt 5.3.2). Um das unmittelbare Ergebnis des Maskenalgorithmus, vom finalen, gemeinsamen Maskenbild begrifflich zu unterscheiden, ist im Text gewöhnlich von einem *vorläufigen* Maskenbild die Rede.

lerdings auch einen universellen Ablauf, der sich für ein vollautomatisches Vorgehen eignet.

### 5.3.1 Wasserscheidentransformation

Das Region Growing (Ausdehnen der Keimzellen) wird als markergesteuerte Wasserscheidentransformation realisiert. Um besser zu verstehen, wie das Maskenbild das Region Growing beeinflusst, ist es im Vorfeld sinnvoll, die Wasserscheidentransformation (WST) näher zu betrachten<sup>1</sup>: In natürlichen Landschaften trennen sogenannte Wasser-

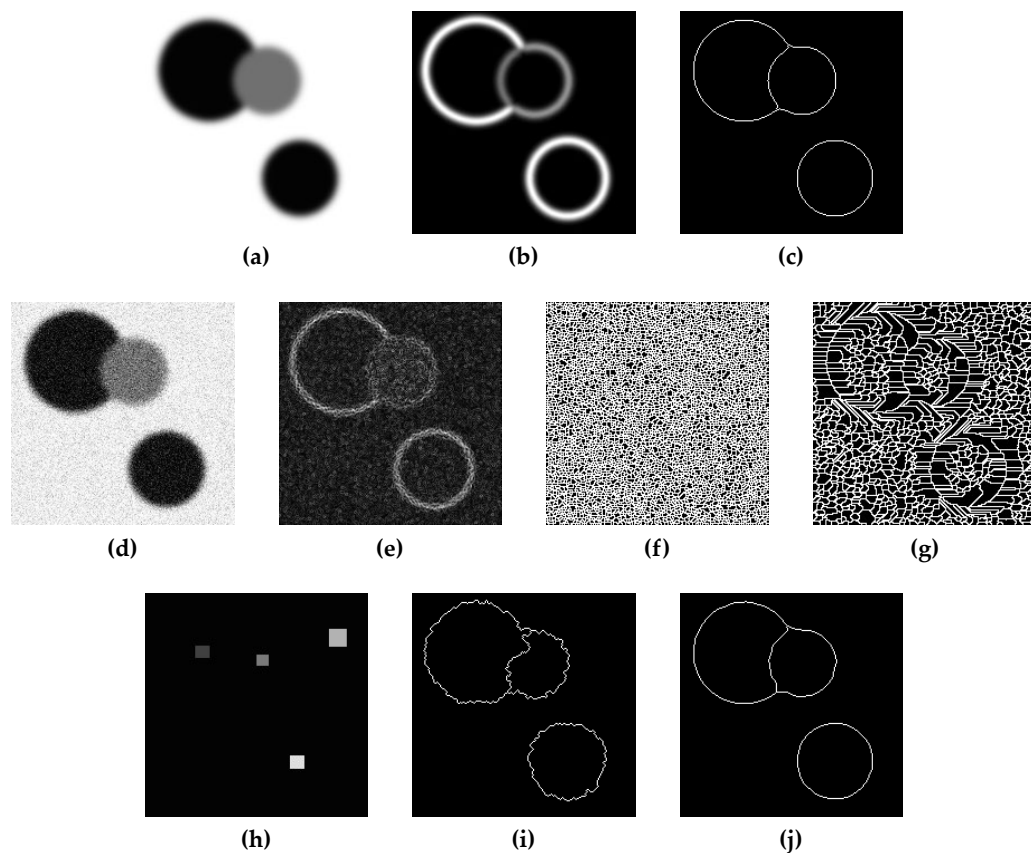
<sup>1</sup>Als Quelle diente primär „*Digital Image Processing*“ (Gonzalez, Woods, 2008) [37]

scheiden die Einzugsgebiete benachbarter Flusssysteme. Davon inspiriert, interpretiert die WST ein Grauwertbild analog zu einem topografischen Relief und unterteilt dieses in Bassins, die durch Wasserscheiden voneinander separiert werden. Fällt ein gedachter Wassertropfen auf eine Wasserscheide, steht im topografischen Model nicht eindeutig fest, in welches der benachbarten Bassins das Wasser fließt. Für alle anderen Bildpunkte ist hingegen eindeutig bestimmt, welchem Einzugsgebiet diese zuzuordnen sind. Eine konkrete Umsetzung der WST bestimmt zumindest eine Unterteilung des Bildes in Bassins, viele gängige Algorithmen bestimmen zusätzlich eine zusammenhängende Grenzlinie zwischen den Bassins, die gesondert gekennzeichnet wird. Sowohl Bassins als auch Wasserscheiden lassen sich durch simuliertes Fluten des Reliefs bestimmen. Dazu wird ein gedachter Grundwasserspiegel angehoben, wobei sich das Wasser in den Senken des Reliefs sammelt. Vereinigen sich dabei die Fluten benachbarter Senken, geschieht dies entlang der gesuchten Wasserscheiden. Es gilt somit, entsprechende Berührungspunkte zu identifizieren. Eine solche Wasserscheidentransformation eignet sich besonders zur Segmentierung von Darstellungen der Gradientenmagnitude (*Gradientenbild*), da sich dieses sinnvoll als topologisches Relief interpretieren lässt<sup>1</sup>. Die WST platziert die Wasserscheiden dabei entlang der Kanten des Ursprungsbildes, auch wenn diese lokal nur schwach ausgeprägt sind. Verrauschte Bilder werden jedoch übersegmentiert, da das zugehörige Gradientenbild viele lokale Minima (Bassins) aufweist. Diese Problematik ist in Abbildung 5.5 deutlich zu sehen.

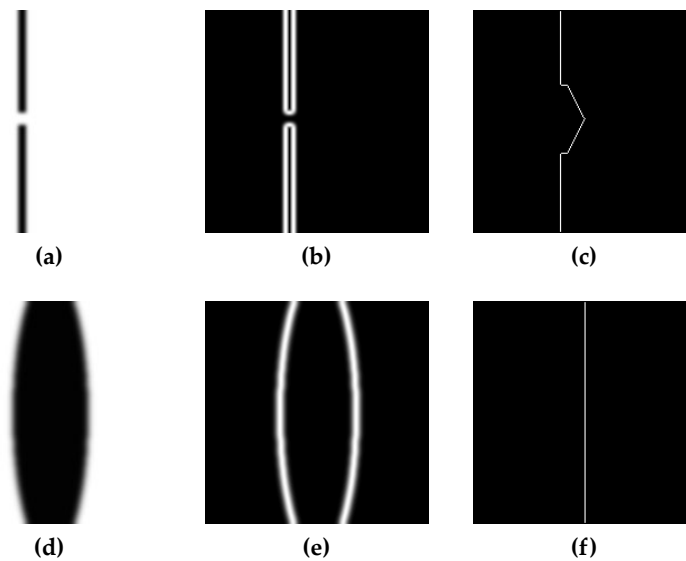
Vorausgesetzt es können einzelne Bildpunkte den Vordergrundobjekten und dem Hintergrund zugeordnet werden, liefert die markergesteuerte Wasserscheidentransformation gute Ergebnisse: Diese entspricht der gewöhnlichen WST, wobei die Senken jedoch ausgehend von den markierten Bildpunkten geflutet werden. In einem separaten Markerbild werden die Bildpunkte dazu numerisch gekennzeichnet; ein reservierter Wert identifiziert dabei unbekannte Bildbereiche. Der angepasste Algorithmus lässt sich wieder mit einem gedachten, steigenden Grundwasserspiegel veranschaulichen, nunmehr ist jedoch nicht das gesamte Relief wasserdurchlässig, sondern das Grundwasser kann nur an den markierten Bildpunkten an die Oberfläche gelangen. Zunächst werden deshalb nur wenige Senken geflutet, allerdings kann die Flut in benachbarte Senken *hinüberschwappen*. Bevor sich die Fluten, die von verschiedenen Markierungen ausgehen, vereinigen, werden analog zur ursprünglichen Wasserscheidentransformationen *Dämme* errichtet. Die Bassins entsprechen wieder von Dämmen eingegrenzten Bildbereichen, allerdings garantiert nicht jede Implementierung, dass die Grenzlinien in allen Fällen den Wasserscheiden des betrachteten Reliefs folgen. Außerdem kann es zu einer Art *Ausbluten* kommen, wenn markierte Bassins von einer vollständig unterbrochenen Kante separiert werden (Abbildung 5.6). Die Grenzlinien werden dennoch als „Wasserscheide“ bezeichnet. Die genannten Ungenauigkeiten treten auf, wenn sich die Flut, ausgehend von den jeweiligen Markierungen, gleichberechtigt, also mit gleicher Geschwindigkeit,

<sup>1</sup>Homogene Bildbereiche werden durch Kanten vom Rest des Bilds abgegrenzt. Kanten entsprechen dabei einer deutlichen Tonwertänderung, wohingegen die Tonwertänderung in homogenen Bildbereichen gering ist. Dementsprechend werden homogene Bildbereiche auf Senken im *Gradientenbild* abgebildet.





**Abbildung 5.5:** Bildsegmentierung mit der Wasserscheidentransformation: Die Wasserscheidentransformation eignet sich gut um unverrauschte Bilder zu segmentieren (obere Reihe). Wird die WST hingegen auf verrauschte Bilder angewandt, sind die Ergebnisse unbrauchbar (mittlere Reihe). Das Rauschen kann mit einem Tiefpassfilter reduziert werden, die Ergebnisse lassen sich bei sinnvollen Filterstärken jedoch nur marginal verbessern (g). Mit der markergesteuerten Wasserscheidentransformation lassen sich hingegen gute Ergebnisse erreichen (untere Reihe). Einzelne Bildpunkte der drei Vordergrundobjekte und des Hintergrunds wurden dazu in einem separaten Markerbild gekennzeichnet (h). (a, d) zeigen das Ausgangsbild, das zugehörige *Gradientenbild* ist in (b, e) dargestellt. Das Resultat der Wasserscheidentransformation ist in (c, f) abgebildet, das der markergesteuerten WST in (i). Für die verrauschten Ausgangsbilder wurde die Wasserscheidentransformation zusätzlich auf den, mit einem Gaußfilter vorverarbeiteten, Bildern durchgeführt (g, j). Die Zwischenschritte sind nicht dargestellt.



**Abbildung 5.6:** Die markergesteuerte Wasserscheidentransformation, angewandt auf die in (a, d) gezeigten Ausgangsbilder: Die Abbildung illustriert mögliche Ungenauigkeiten der markergesteuerten WST (siehe Text). Es wurden jeweils der linke und rechte Bildrand markiert. Im Ergebnis folgt die Abgrenzung der markierten Bassins nicht den Objektgrenzen (c, f). (b, e) zeigen das zugehörige Gradientenbild.

ausbreitet. Eine mögliche Alternative bestünde darin, die Markierungen der Reihe nach zu betrachten und dabei sofort alle erreichbaren Bildbereiche (entsprechend der Höhe der Flut), die noch nicht anderen Markierungen zugeordnet wurden, zu fluten. Das Ergebnis wäre dann allerdings abhängig von der Reihenfolge, in der die Markierungen betrachtet würden. Auch wäre das Ergebnis eines solchen Algorithmus bei mehrfacher Kennzeichnung desselben Bassins womöglich unerwartet. Verdeutlichen lässt sich dies anhand der Segmentierung eines leeren Gradientenbilds: Werden alle Markierungen gleichberechtigt behandelt, entstehen um die Markierungen herum Voronoizellen, alle Bildpunkte werden also der nächstgelegenen Markierung zugewiesen. Im anderen Fall würde der gesamte Bildbereich, abgesehen von den explizit markierten Bildpunkten, der zuerst betrachteten Markierung zugeordnet.

*MSeam* nutzt die markergesteuerte Wasserscheidentransformation zur Umsetzung des Region Growing. Die Keimzellen entsprechen dabei Markierungen am Rand des zu segmentierenden Bildbereichs. Dabei ist es nicht ausgeschlossen, dass eine Senke des betrachteten Reliefs mehrfach gekennzeichnet wird, alle Markierungen sollten deshalb gleichberechtigt behandelt werden. Daraus ergibt sich auch die Möglichkeit mit *MSeam* eine Voronoisegmentierung durchzuführen (Abschnitt 2) - die markergesteuerte WST muss dazu lediglich (mit gesetzten *Markern*) auf ein leeres Bild (ebenes Relief) angewandt werden.

Die (markergesteuerte) Wasserscheidentransformation kann vielfältig implementiert werden. Naheliegend ist ein simuliertes Fluten entsprechend obiger Betrachtung. Insbesondere die markergesteuerte Wasserscheidentransformation lässt sich auf diese Weise mit linearer Komplexität umsetzen. Im Falle der gewöhnlichen WST besteht ein Basin aus allen Punkten, die diesem entsprechend geodätischer Distanz (Distanz unter Berücksichtigung der Topografie) am Nächsten liegen. Mit einer geeigneten diskreten Approximation der geodätischen Distanz, sind weitere bekannte Implementierungen möglich. Nicht alle Umsetzungen des Verfahrens entsprechen vollständig obiger Betrachtung. *MSeam* kann für die Nutzung unterschiedlicher Verfahren konfiguriert werden. Empfehlenswert sind Implementierungen, die der vorangegangenen Betrachtung der Wasserscheidentransformation entsprechen.

Gängige Varianten der WST kennzeichnen die Wasserscheiden explizit. Im Kontext von *MSeam* ist es jedoch erforderlich, auch die Wasserscheiden eindeutig den jeweiligen Keimzellen zuzuordnen. Im einfachsten Fall erfolgt die Zuordnung willkürlich nach dem numerischen Wert der angrenzenden Marker. Die Zuweisung kann aber auch auf Grundlage von Bildmerkmalen erfolgen, etwa dergestalt, dass die Tonwertdifferenz entlang der resultierenden Nahtlinie minimiert wird.

### 5.3.2 Berechnung des Maskenbilds

Ein einkanaliges Grauwertbild, das sogenannte Maskenbild, steuert das Region Growing. Die Grauwerte des Maskenbilds geben dabei die Salienz des jeweiligen Bildpunkts an. Entsprechend der vorangegangenen Betrachtung der Wasserscheidentransformation, muss das Maskenbild sinnvoll als Relief interpretiert werden können, was insbesondere auf Gradientenbilder (punktweise Darstellung der Magnitude des diskret approximierten Gradienten) zutrifft. Erschwert wird die Bestimmung des Maskenbilds dadurch, dass dieses mehrere Szenen angemessen berücksichtigen und dabei insbesondere geometrische, radiometrische sowie inhaltliche Abweichung tolerieren muss. Die Erzeugung des Maskenbilds erfolgt in einem fünfstufigen Prozess (Abbildung 5.4):

1. Vorfilter
2. Maskenalgorithmus
3. Maskenfilter
4. punktweises Minimum
5. Nachfilter

*MSeam* ist nicht auf eine bestimmte Ausgestaltung dieses Vorgehens angewiesen, vielmehr kann der Prozess der Berechnung des gemeinsamen Maskenbilds flexibel kon-

figuriert<sup>1</sup> (XML-Projektdatei) und per *Python*-API erweitert werden. Ohne spezifische Konfiguration nutzt *MSeam* einen Standardablauf, der für die vollautomatische Mosaikierung geeignet ist.

Die Berechnung des Maskenbilds erfolgt beschränkt auf den Ausschnitt der aktuell betrachteten Überdeckung. *MSeam* macht sich bei der Berechnung die Kommutativität und Assoziativität der Minimum-Operation zu Nutze und verwaltet das Ergebnis von Schritt (4) in einem zweistufigen *In-Memory*- und *On-Disk-Cache*. Soll nun das Maskenbild für eine Überdeckung des Grads drei oder höher berechnet werden, ist es möglich, Zwischenergebnisse vorangegangener Compositing-Iterationen (die umgebenden Überdeckungen niedrigeren Grads wurden bereits in früheren Iterationen betrachtet), soweit verfügbar, weiterzuverwenden.

**1) Vorfilter** Der Vorfilter ist geeignet, um Rauschen und feinste Strukturen zu unterdrücken, etwa mit einem Gaußfilter. Darüberhinaus lässt sich im Vorfilterschritt eine Vorsegmentierung der Bilddaten durchzuführen. Dazu können Clustering-Algorithmen wie *K-Means* oder morphologische *Quasi-Flatzone*-Filter<sup>2</sup> eingesetzt werden. Weitestgehend unabhängig vom Maskenalgorithmus bestimmt der Vorfilter, ob die Nahtlinie weiträumig ausgedehnten Strukturen folgt, oder bevorzugt feinen Details (einschließlich Rauschen und Strukturen der Größenordnung einzelner Pixel). Darüber hinaus ließe sich der Vorfilter auch nutzen, um die Bildausschnitte geometrisch zu skalieren: Bei geeignetem Vorgehen würden ebenfalls Details unterdrückt und gleichzeitig das Datenvolumen reduziert. Diese Möglichkeit ist jedoch bisher in der Implementierung von *MSeam* nicht vorgesehen.

**2) Maskenalgorithmus** Berechnung eines *vorläufigen* Maskenbilds für jeden der Ausschnitte. Auch für mehrkanalige Ausgangsbilder muss das Maskenbild einer skalaren Funktion entsprechen. Das Maskenbild darf also nur einen Kanal besitzen. Gut eignet sich etwa die Bestimmung einer approximierten Gradientenmagnitude für jeden Bildpunkt (*Gradientenbild*), da sich das Ergebnis sinnvoll als (topografisches) Relief auffassen lässt, wobei die Höhe des Reliefs das Kantenpotential als Salienz Kriterium angibt. Es gibt mehrere Möglichkeiten die Gradientenmagnitude zu bestimmen, *Morphological Image Compositing* verwendet zum Beispiel einen sogenannten morphologischen Gradienten (Formel 4.9) oder ein entsprechendes Verfahren für multispektrale Bilder (Formel 4.10). *MSeam* nutzt hingegen standardmäßig den sogenannten *Scharr*-Operator um den Gradienten lokal zu approximieren. Dies geschieht durch Faltung jedes Bildes mit einem 3x3-Faltungskern (Abbildung 5.7).

<sup>1</sup>Dabei wurde auch die Notwendigkeit berücksichtigt, innerhalb eines Filterschritts gegebenenfalls mehrere Filter zu kombinieren beziehungsweise hintereinander auszuführen.

<sup>2</sup>Bestimmung von Gruppen ähnlicher Pixel, die eine festgelegte Mindestgröße überschreiten. Im weiteren Verlauf werden die Gruppen unter Berücksichtigung eines Ähnlichkeitsmaßes um angrenzende Pixel erweitert, bis keine weiteren Bildpunkte verbleiben. Ausgehend von den so bestimmten Gruppen wird das Bild vereinfacht: Über alle Pixel einer Gruppe wird zum Beispiel der durchschnittliche Pixelwert bestimmt und allen Bildpunkten dieser Gruppe zugewiesen.

Die Gradientenbeträge werden daraus für panchromatische (einkanalige) Szenen mit der euklidischen Norm bestimmt. Es ist naheliegend, diesen Ansatz auf mehrkanalige Bilder zu erweitern, in dem für jeden Kanal zunächst unabhängig ein Gradientenbild berechnet wird. Die vorläufigen Gradientenbilder können dann etwa durch Bestimmung des punktweisen Maximums oder Bildung des Durchschnitts zu einem gemeinsamen Maskenbild vereint werden. Dabei wird implizit vorausgesetzt, dass die Kanäle zu einem gewissen Grad übereinstimmen, was jedoch nicht allgemein angenommen werden kann. Alternativ bietet es sich an, multispektrale Aufnahmen als Vektorfeld aufzufassen und anhand dessen die Rate der maximalen Änderung zu bestimmen. *MSeams* Implementierung dieses Ansatzes folgt dabei dem Vorschlag von Di Zenzo [37], [38] (siehe auch Abschnitt 5.4): Seien zunächst  $F_i(x, y)$  ein Bildkanal,  $g_{xx}$  und  $g_{yy}$  die Norm der Gradienten entlang  $x$ - und  $y$ -Achse sowie  $g_{xy}$  das Skalarprodukt der Gradienten.

$$g_{xx} = \sum_{\forall i} \|\partial_x F_i\|^2 \quad (5.1)$$

$$g_{yy} = \sum_{\forall i} \|\partial_y F_i\|^2 \quad (5.2)$$

$$g_{xy} = \sum_{\forall i} \partial_x F_i \partial_y F_i \quad (5.3)$$

Di Zenzo leitet daraus die Richtung der maximalen Änderung

$$\Theta_C(x, y) = \frac{1}{2} \tan^{-1} \left[ \frac{2g_{xy}}{g_{xx} - g_{yy}} \right] \quad (5.4)$$

ab. Wegen der Periodizität der Tangensfunktion existieren neben der Lösung  $\Theta_{C0}$  noch weitere Lösungen  $\Theta_{C0} + n\pi/2$ . Es reicht, sich auf einen Intervall, etwa  $[0, \pi)$  zu beschränken. Die Rate der maximalen Änderung  $C(x, y)$  ergibt sich dann wie folgt:

$$C(x, y) = \max_{\theta = \Theta_{C0} + n\pi/2} C_\theta(x, y) \quad (5.5)$$

$$C_\theta(x, y) = \left\{ \frac{1}{2} [(g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos 2\theta(x, y) + 2g_{xy} \sin 2\theta(x, y)] \right\}^{1/2} \quad (5.6)$$

Bei dem (vorläufigen) Maskenbild muss es sich nicht zwangsläufig um ein Gradientenbild handeln. Alle Darstellungen eines Salienzkriteriums, die sich im Sinne der Wasserscheidentransformation interpretieren lassen, sind geeignet. Naheliegend ist die Verwendung eines fortgeschritteneren Verfahrens zur Kantendetektion, etwa des sogenannten *Canny*-Kantendetektors. Das Ergebnis solcher Algorithmen ist dabei üblicherweise eine binäre (*flache*) Kennzeichnung der Kantenverläufe, es ist aber auch möglich, die Kanten entsprechend ihrer Ausdehnung und weiterer Kriterien wie Kompaktheit (langgestreckte Kantenzüge sollen bevorzugt werden)

$$S_h = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \quad S_v = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix}$$

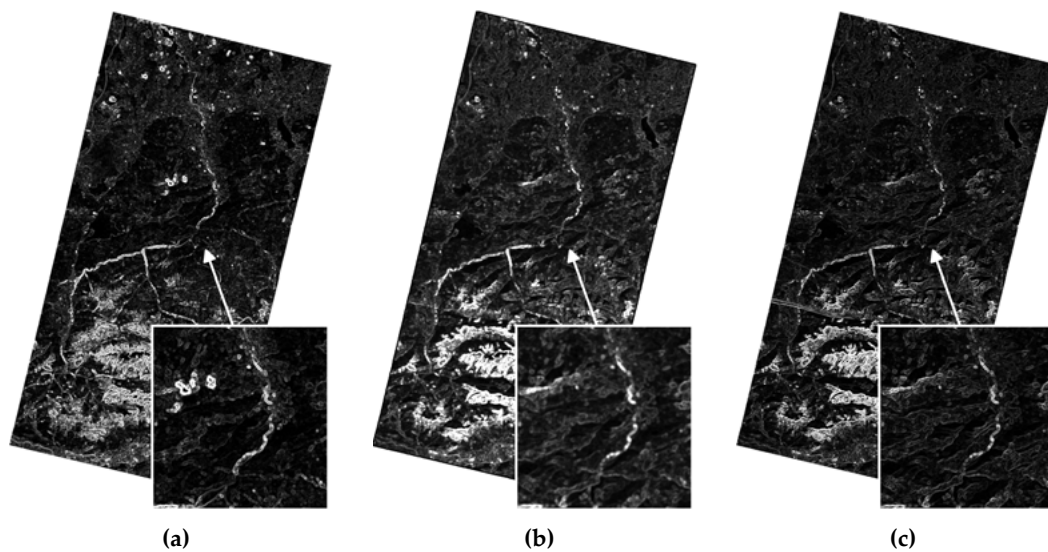
**Abbildung 5.7:** Scharr-Operator (3x3-Faltungskern) zur approximativen Bestimmung des horizontalen ( $S_h$ ) und vertikalen ( $S_v$ ) Gradienten. Der gezeigte Faltungskern ist separierbar.

zu kennzeichnen. Ein Beispiel ist in Abbildung 5.9 abgebildet. *MSeam* bietet auch Algorithmen für dieses und ähnliche Verfahren.

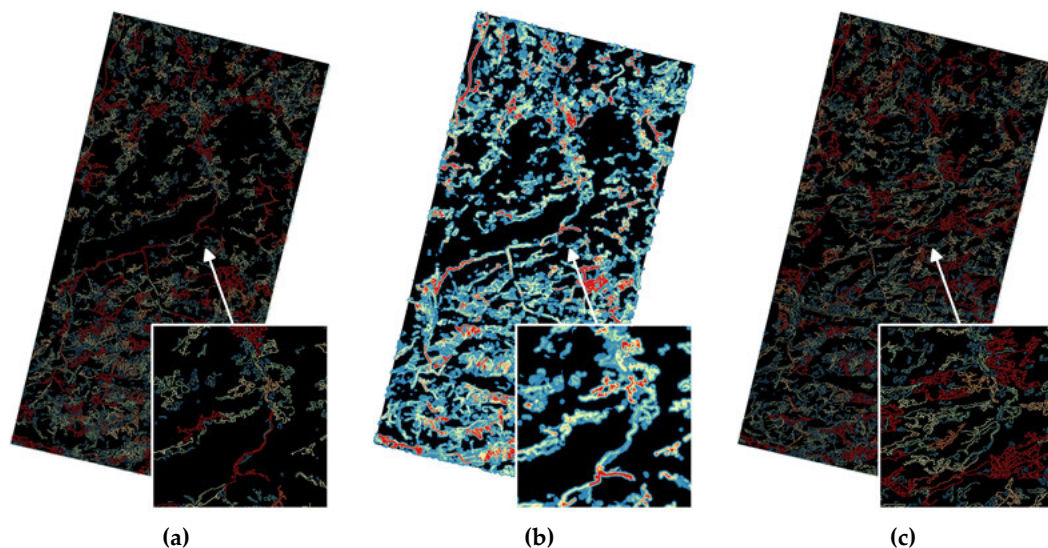
**3) Maskenfilter** Die sich überlappenden Bildbereiche sind nicht identisch. Insbesondere sind geometrische Abweichungen in Folge einer unpräzisen Ausrichtung der Einzelaufnahmen möglich. Auch verändern sich natürliche Geländestrukturen wie Flüsse und Vegetationsgrenzen im Laufe der Zeit, stellen aber dennoch eine gute Wahl für die Nahtlinien dar. Somit sind die interessanten salienten Strukturen (Kantenmerkmale) der beitragenden Orthofotos im Allgemeinen geringfügig gegeneinander verschoben. Um solche Geländemerkmale dennoch zu berücksichtigen, muss das Salienzmerkmal auch die Umgebung eines Bildpunkts berücksichtigen und zum Beispiel Kantenmerkmale nur unscharf abbilden. Andernfalls würde ein guter Teil der Merkmale bei der nachfolgenden Bildung des punktwisen Minimums ausgeschlossen werden, die markergesteuerte Wasserscheidentransformation tendiert dann lokal zu einer Voronoisegmentierung (Abschnitt 2). Die Wahl des Maskenalgorithmus wird dadurch aber nur geringfügig eingeschränkt, denn auch durch ein nachträgliches Weichzeichnen des Maskenbilds können angrenzende Bildpunkte in das Salienzmerkmal einbezogen werden, wobei sich die Größe der Umgebung gut kontrollieren lässt. Bewährt hat sich diesbezüglich ein Gaußfilter.

Zusätzlich kann es sinnvoll sein, das *vorläufige* Maskenbild anhand zusätzlicher Kriterien zu skalieren (die Grauwerte), etwa wenn eine Priorisierung langgestreckter (*zielstrebigere*) Kantenzüge bei der Platzierung der Seamlines gewünscht ist. Die Implementierung von *MSeam* enthält zwei entsprechende Maskenfilter. Diese verwenden den *Canny*-Kantendetektor [37] und ein Kompaktheitskriterium oder den sogenannten *Line Segment Detector* [39] um ausgedehnte Kantenzüge zu identifizieren. Durch geeignetes Skalieren des *vorläufigen* Maskenbilds wird anschließend die Umgebung solcher Kanten besonders gewichtet.

**4) Punktwises Minimum** Zwar trennen die Nahtlinien jeweils Paare von Szenen, *Morphological Image Compositing* und *MSeam* beschränken sich jedoch auf Merkmale, die allen Szenen einer Überdeckung gemein sind. Somit ist es möglich, die Keimzellen in einem unkomplizierten Region-Growing-Schritt auszudehnen (markerge-



**Abbildung 5.8:** Maskenbild einer paarweisen Überlappung zweier Multispektralaufnahmen, berechnet gemäß der Konfiguration für das vollautomatische Mosaikieren (Abschnitt 5.4). In (a) und (c) werden die, zunächst unabhängig berechneten, *vorläufigen* Maskenbilder gezeigt. Diese wurden in einem anschließenden Maskenfilter-Schritt weichgezeichnet um dann das eigentliche Maskenbild (b) als punktweises Minimum zu bestimmen.



**Abbildung 5.9:** Maskenbild einer paarweisen Überlappung zweier Multispektralaufnahmen; (b) ist das eigentliche Maskenbild, (a, c) zeigen das Resultat des verwendeten Kantendetektors. Die Berechnung erfolgte mit dem „WeightedCannyMaskAlgorithm“: Die Hysterese-Parameter des zugrundeliegenden *Canny*-Kantendetektors werden dabei in Abhängigkeit von den Bildinhalten bestimmt; die gefundenen Kanten werden nach den Kriterien Fläche und Kompaktheit (langgestreckte, also weniger kompakte Kanten sind besser) gewichtet. Im Maskenfilterschritt wurden die Kantenlabel verbreitert (Maximum-Filter) und anschließend weichgezeichnet. (b) wurde wiederum als punktweises Minimum berechnet.

steuerte WST). Entsprechend repräsentiert das Maskenbild die jeweils minimale Salienz aller relevanten Szenen.

**5) Nachfilter** In einem abschließenden Nachfilter-Schritt ist es möglich, Kriterien einzubeziehen, die eine gemeinsame Betrachtung aller Szenen eines Überlappungsbereichs erforderlich machen, zum Beispiel, die Berücksichtigung radiometrischer Kriterien: Dazu wird für jeden Bildpunkt innerhalb der betrachteten Überdeckung, die maximale, paarweise Tonwertdifferenz bestimmt und das Relief des Maskenbilds entsprechend skaliert.

## 5.4 Vollautomatisches Mosaikieren

Das gewählte Salienzmerkmal und die Aufbereitung des zugehörigen Maskenbilds bestimmen maßgeblich das Ergebnis von *MSeam*. Somit ist es naheliegend den Prozess der Erzeugung des Maskenbilds flexibel konfigurierbar zu gestalten. Letztlich wird aber ein vollautomatisches Vorgehen angestrebt, dem eine Vielzahl von Parametern grundsätzlich entgegensteht. Es gilt somit einen Prozess zur Berechnung des Maskenbilds festzulegen, der universell einsetzbar ist.

Es ist sinnvoll als Erstes einen Maskenalgorithmus festzulegen, da dieser das finale Maskenbild maßgeblich beeinflusst. Kantendetektoren wurden dafür von vornherein ausgeschlossen, da die binäre Kennzeichnung von Kanten die Salienz von Bildpunkten nur eingeschränkt widerspiegelt und in Kombination mit der markergesteuerten Wasserscheidentransformation Artefakte (Abbildung 5.6) begünstigt. Insbesondere entspricht es der Natur eines solchen binären Merkmals, dass nur einige, ausgezeichnete Pixel markiert werden (*dünnbesetztes* Maskenbild). Da nun die gefundenen Kantenzüge nicht notwendigerweise zusammenhängend verlaufen, müssen die einzelnen Abschnitte bei der Platzierung der Nahtlinien verbunden werden. Ein dünnbesetztes Maskenbild bietet für die Zwischenbereiche jedoch keine Anhaltspunkte.

Das Maskenbild muss mehrere überlappende Ausgangsbilder angemessen berücksichtigen, wobei die interessanten salienten Strukturen (etwa Kantenmerkmale) im Allgemeinen geringfügig gegeneinander verschoben sind. In der Praxis ist es daher nicht möglich, die Seamline nur anhand von Merkmalen zu platzieren, die exakt übereinstimmen. Stattdessen wird eine ungefähre Wahl der Nahtlinie angestrebt. Dazu darf das Maskenbild saliente Merkmale nur unscharf wiedergeben, wobei für jeden Bildpunkt auch eine gewisse Umgebung zu berücksichtigen ist. In *Morphological Image Compositing* wird dementsprechend ein sogenannter morphologischer Gradient (Formel 4.9) genutzt, der die maximale (absolute) Differenz der Grauwerte in einer festgelegten Umgebung bestimmt und als Salienzmerkmal verwendet. Wird dabei eine großflächige Umgebung berücksichtigt, werden lokale Details vollständig ausgeblendet, obwohl diese zu einer



besseren Platzierung der Nahtlinie beitragen können. Auch lässt sich dieses Verfahren nur eingeschränkt auf Mehrkanalbilder erweitern (Abschnitt 4.10), da eine effiziente Implementierung das Vorhandensein eines maximalen und minimalen Pixelwerts erforderlich macht, für Vektoren aber keine entsprechende Ordnungsrelation definiert ist. Um diesem Problem zu entgehen, sieht *MSeam* für das vollautomatische Mosaikieren die Verwendung des *Scharr*-Operators (Abbildung 5.7) vor, wobei im panchromatischen Fall der Gradientenbetrag (euklidische Norm) und für Mehrkanalbilder die Rate der maximalen Änderung nach *Di Zenzo* (Abschnitt 5.3.2) als Salienzmerkmal genutzt wird. Dabei wird nur eine sehr kleine Umgebung jedes Bildpunkts berücksichtigt, weshalb saliente Strukturen im berechneten *Gradientenbild* scharf wiedergegeben werden. Im Maskenfilter-Schritt wird aus diesem Grund ein Gaußfilter eingesetzt um die Umgebung einzubeziehen, wobei sich ein Filterkernel der Größe  $7 \times 7$  ( $\sigma = 1,4$ ) bewährt hat. Auf eine weitergehende Nutzung des Maskenfilters (etwa eine Skalierung des Maskenbilds) wird verzichtet. Ebenso wird kein Nachfilter (Abschnitt 5.3.2) eingesetzt.

Es verbleibt die Wahl eines geeigneten Vorfilters, um feine Details und gegebenenfalls Rauschen auszufiltern. Dazu bietet sich wiederum ein Gaußfilter an. In den Experimenten hat sich ein Filterkernel der Größe  $5 \times 5$  ( $\sigma = 1,1$ ) als gute und universelle Wahl erwiesen. Um eine skaleninvariante Platzierung der Seamline zu erhalten, müsste der Filterkernel analog zur (physikalischen) Auflösung skaliert werden. Bei der vollautomatischen Mosaikierung soll jedoch der verfügbare Detailgrad bestmöglich ausgeschöpft werden. Der Vorfilter soll sich somit auf das Glätten von Rauschen und, insbesondere, Strukturen der Größenordnung einzelner Pixel (unabhängig von der physikalischen Auflösung) beschränken. Für diesen Zweck reicht es aus, einen universellen Vorfilter, unabhängig von der Auflösung, festzulegen.

## 5.5 Maskieren unerwünschter Bildausschnitte

Häufig ist es von besonderem Interesse, Ausschnitte einzelner Szenen von der Mosaikierung auszuschließen, also Bildbereiche zu maskieren. Ist dies gewünscht, muss für die betroffenen Szenen individuell eine entsprechende Polygonmaske erstellt werden. Da *MSeam* keine Anforderungen an die Form der Definitionsbereiche einzelner Szenen stellt, reicht es dabei aus, die maskierten Bereiche (*Maske*; nicht zu verwechseln mit *Maskenbild*) vom Definitionsbereich einer Szene abzuziehen, ganz als ob in diesem Bereich keine Daten zur Verfügung stünden. Tatsächlich verwaltet *MSeam* den Definitionsbereich unabhängig von einer möglicherweise vorhanden Maske und verrechnet diese nicht von vornherein. Dadurch wird es möglich Teile (einzelne Polygone) einer Maske zu identifizieren, die nicht von mindestens einer weiteren Szene vollständig überdeckt werden. In diesen Bereichen würde das finale Mosaik *Löcher* aufweisen. Tatsächlich, ist es aber häufig sinnvoller innerhalb solcher *Löcher*, den eigentlich maskierten Bereich (zum Beispiel eine Wolke) beizubehalten. Dies wird erreicht, indem gegebenenfalls einzelne Polygone der Maske ignoriert werden. Der verbleibende Teil der Maske (effektive Mas-

ke) wird dann vom Definitionsbereich der zugehörigen Szene abgezogen (Bestimmung des effektiven Definitionsbereichs). Das geschieht vor dem Identifizieren der Überlappungsbereiche, da diese vom effektiven Definitionsbereich abhängen. Eine Änderung der Maske einer Einzelaufnahme lässt sich einfach berücksichtigen indem die betroffene Szene zunächst aus dem Mosaik entfernt wird, um anschließend mit der aktualisierten Maske wieder hinzugefügt zu werden.

## 5.6 Renderschritt und Filtern von Artefakten

In den Compositing-Iterationen wurden die Szenen auf angrenzende Überdeckungen ausgedehnt. Im Zuge dessen ließen sich die finalen Beiträge der Szenen zum Mosaik prinzipiell inkrementell zusammensetzen, worauf jedoch zugunsten eines dedizierten Renderschritts verzichtet wird. Dieses Vorgehen ist bei der iterativen Erweiterung eines Mosaiks vorteilhaft und bietet darüberhinaus die Möglichkeit, Artefakte zu erkennen und auszufiltern.

Das *Rendern* erfolgt, begleitend zu einer optionalen Ausgabe des Mosaiks in Form von Raster- oder Vektordaten. Zu diesem Zeitpunkt ist die Unterteilung aller Überdeckungen in die Anteile angrenzender Szenen bereits bekannt. Dementsprechend stehen auch die (disjunkten) finalen Beiträge der Szenen zum Mosaik eindeutig fest. Diese können unmittelbar aus den gespeicherten Zwischenergebnissen abgeleitet werden. Dabei gilt es, die geschachtelte Struktur der Überdeckungen zu berücksichtigen: Die gespeicherten Zwischenergebnisse repräsentieren jeweils die gesamte Überdeckung, das Ergebnis ist jedoch nur im Gültigkeitsbereich (Formel 4.7) eindeutig. Ausgehend vom Definitionsbereich der jeweiligen Szene, wird schließlich der finale Beitrag der Szene zum Mosaik bestimmt, indem alle Bereiche, die in den Compositing-Iterationen eindeutig benachbarten Szenen zugeordnet wurden, vom ursprünglichen Definitionsbereich abgezogen werden. Dabei gilt es, alle relevanten Überdeckungen, absteigend nach dem Überdeckungsgrad geordnet, zu betrachten. Innerhalb der Überdeckungen des jeweils höchsten Grads ist das Ergebnis zwangsläufig eindeutig und kann direkt vom ursprünglichen Definitionsbereich der Szene abgezogen werden. Für alle weiteren Überdeckungen lässt sich der Gültigkeitsbereich bei einem solchen geordneten Vorgehen sehr einfach bestimmen indem die Bereiche bereits betrachteter Überdeckungen ignoriert beziehungsweise abgezogen werden.

Ausgehend von den Keimzellen, werden die Überdeckungen in der Compositing-Iteration unter den angrenzenden Szenen aufgeteilt. Die Kennzeichnung der Keimzellen legt dabei fest, welche Szene durch die jeweilige Keimzelle repräsentiert wird. Keimzellen, die zu einer bestimmten Szene gehören, müssen im Allgemeinen nicht zusammenhängend entlang des Rands einer Überdeckung platziert sein, die genaue Verteilung der Keimzellen hängt dabei von der Aufteilung der umgebenen Bildbereiche und der Form der Überdeckung ab. Im Zuge des Region Growing werden die Keimzellen in die Über-

deckung hinein ausgedehnt. Dabei entsteht für jeden Bereich zusammenhängender und identisch gekennzeichnete Keimzellen wieder ein zusammenhängender Bereich. Die Bereiche, die aus identisch gekennzeichneten Keimzellen hervorgegangen sind, können sich grundsätzlich vereinigen. Ob und wo dies geschieht, hängt dabei vom Maskenbild ab. Dementsprechend kann es passieren, dass insbesondere bei Überdeckungen höheren Grades Artefakte entstehen. Das sind isolierte Bereiche gleicher Kennzeichnung, die nicht zusammenhängen. Ein solches Szenario ist in Abbildung 5.10 abgebildet.

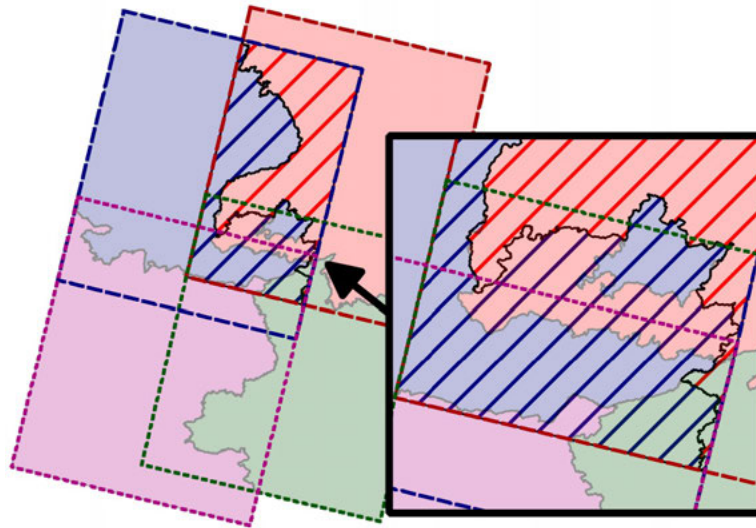
Derartige „Inseln“ sind unerwünscht, da sie vermeidbare Nahtlinien nach sich ziehen. *MSeam* kann diese Artefakte optional unterdrücken beziehungsweise nachträglich filtern. Dabei wird vorausgesetzt, dass die ungefilterten Beiträge aller Szenen innerhalb der betrachteten Umgebung feststehen, also bereits bestimmt wurden. Zunächst wird festgestellt, ob der ungefilterte Beitrag der momentan betrachteten Szene aus mehreren, nicht zusammenhängenden Bereichen besteht, ob also überhaupt Artefakte vorhanden sind. Anschließend werden identifizierte Artefakte, wenn möglich, angrenzenden Beiträgen anderer Szenen zugeordnet. Eine solche Zuordnung ist immer dann möglich, wenn dabei keine neuen Artefakte entstehen. Ein geeigneter (ungefilterter) angrenzender Beitrag muss das Artefakt somit zumindest teilweise berühren. Auch ist es zwingend notwendig, dass die Szene, die den angrenzenden Beitrag beisteuert (genauer deren effektiver Definitionsbereich), auch den Bereich des betrachteten Artefakts umfasst. Eine solche Zuordnung ist eindeutig möglich, wenn das Artefakt nur an einen weiteren Beitrag angrenzt. Andernfalls muss eine geeignete Auswahl getroffen werden. Dazu bietet es sich beispielsweise an, die Länge des gemeinsamen Rands zu betrachten<sup>1</sup>.

## 5.7 Vor- und Nachteile: Vergleich mit *Morphological Image Compositing*

*MSeam* baut auf *Morphological Image Compositing* auf und erbt damit auch dessen Vor- und Nachteile. Ein Unterscheidungsmerkmal ist der rege Gebrauch von Vektordaten in *MSeam*. Diese lassen sich effizient in gängigen Datenbanksystemen verwalten. Auch das finale Mosaik beziehungsweise dessen Aufteilung in die Beiträge der beteiligten Szenen wird vektoriell beschrieben. Aus den vektoriell gespeicherten Zwischenergebnissen (die Unterteilung der Überlappungsbereiche) lassen sich die Beiträge einfach ableiten und können zum Beispiel als Datenbank-View zur Verfügung gestellt werden. Darauf aufbauend wurde das iterative Mosaikieren implementiert.

*Mseam* stellt keine Anforderungen an die Form der Definitionsbereiche. Es können daher beliebige Bildausschnitte durch Maskierung von der Betrachtung ausgeschlossen werden.

<sup>1</sup>Die bestehende Implementierung berücksichtigt eine, den Szenen zugewiesene, eindeutige ID. Dieses Kriterium ist allerdings abhängig von der Reihenfolge, in der die IDs vergeben werden.



**Abbildung 5.10:** Gezeigt sind die, bereits vollständig berechneten, disjunkten Beiträge von vier Szenen zu einem Mosaik. Der schraffiert dargestellte Bereich entspricht dem 2er-Überlappungsbereich, der von der roten und blauen Szene gebildet wird. Die unterschiedliche Färbung der Schraffur kennzeichnet das Ergebnis der Unterteilung ebendieser Überdeckung (gefundene Nahtlinie schwarz eingezeichnet). Der Beitrag der blauen Szene zum finalen Mosaik ist nicht zusammenhängend: Innerhalb des roten Beitrags (oben rechts) befindet sich eine blaue *Insel* (Artefakt). Dieses Artefakt liegt vollständig innerhalb des Beitrags der roten Szene (daher der Ausdruck *Insel*) und könnte der roten Szene zugeordnet werden (insbesondere liegt die *Insel* auch innerhalb des Definitionsbereichs der roten Szene). Das Artefakt entsteht bei der Aufteilung der [■ ■ ■]-Überdeckung (grün, rot, blau): Die Abgrenzung dieses Überlappungsbereichs zum Gültigkeitsbereich der schraffierten 2er-Überdeckung entspricht der grünen (■) Linie. Die Keimzellen für das *Region Growing* ergeben sich entlang dieser Linie entsprechend der verschiedenfarbigen Schraffur der 2er-Überdeckung. Es gibt somit zwei getrennte Bereiche mit blauen (■) Keimzellen. Die rechts platzierten, blauen Keimzellen sind zu beiden Seiten von roten (■) Keimzellen umgeben. Im gezeigten Beispiel reichte die Geschwindigkeit, mit der die blauen Keimzellen ausgedehnt wurden nicht aus, um die anfänglich getrennten (blauen) Keimzellen zu vereinigen. Die Geschwindigkeit wird durch das zugrundeliegende Maskenbild bestimmt und ist somit von den Bildinhalten abhängig.

Das gilt prinzipiell auch für *Morphological Image Compositing*. Allerdings berücksichtigt dieses die Maske nicht schon bei der Identifizierung der Überlappungsbereiche sondern erst in einem späteren Schritt, was die Bestimmung geeigneter Keimzellen verkompliziert. Die sich daraus ergebenden Sonderfälle wurden in Abschnitt 4 nicht beschrieben (Beschreibung: siehe [22]).

In Abschnitt 4.3 wurden mitunter unpräzise Nahtlinien kritisiert. *MSeam* begegnet dem durch einen flexibel anpassbaren Prozess zur Erzeugung des Maskenbilds (welches das Region Growing maßgeblich beeinflusst). Je nach Konfiguration kann es auch bei *MSeam* zu solchen unpräzisen Nahtlinien kommen. Dies kann jedoch in *MSeam* insbesondere durch Verwendung von auf Kantendetektoren aufbauenden Maskenfiltern reduziert werden. Diese sind nicht Bestandteil des Standardprozesses für die vollautomatische Mosaikierung. Aber auch dieser Universalablauf verwendet an Stelle des morphologischen Gradienten eine, insbesondere für Multispektralaufnahmen, geeignetere Methode zur Bestimmung der Salienz eines Bildpunkts (*Di Zenzo*).

Beiden Verfahren ist gemein, dass prinzipiell isolierte Bereiche (Artefakte) auftreten können. *MSeam* kann diese bei der Berechnung der finalen Beiträge einer Szene ausfiltern.



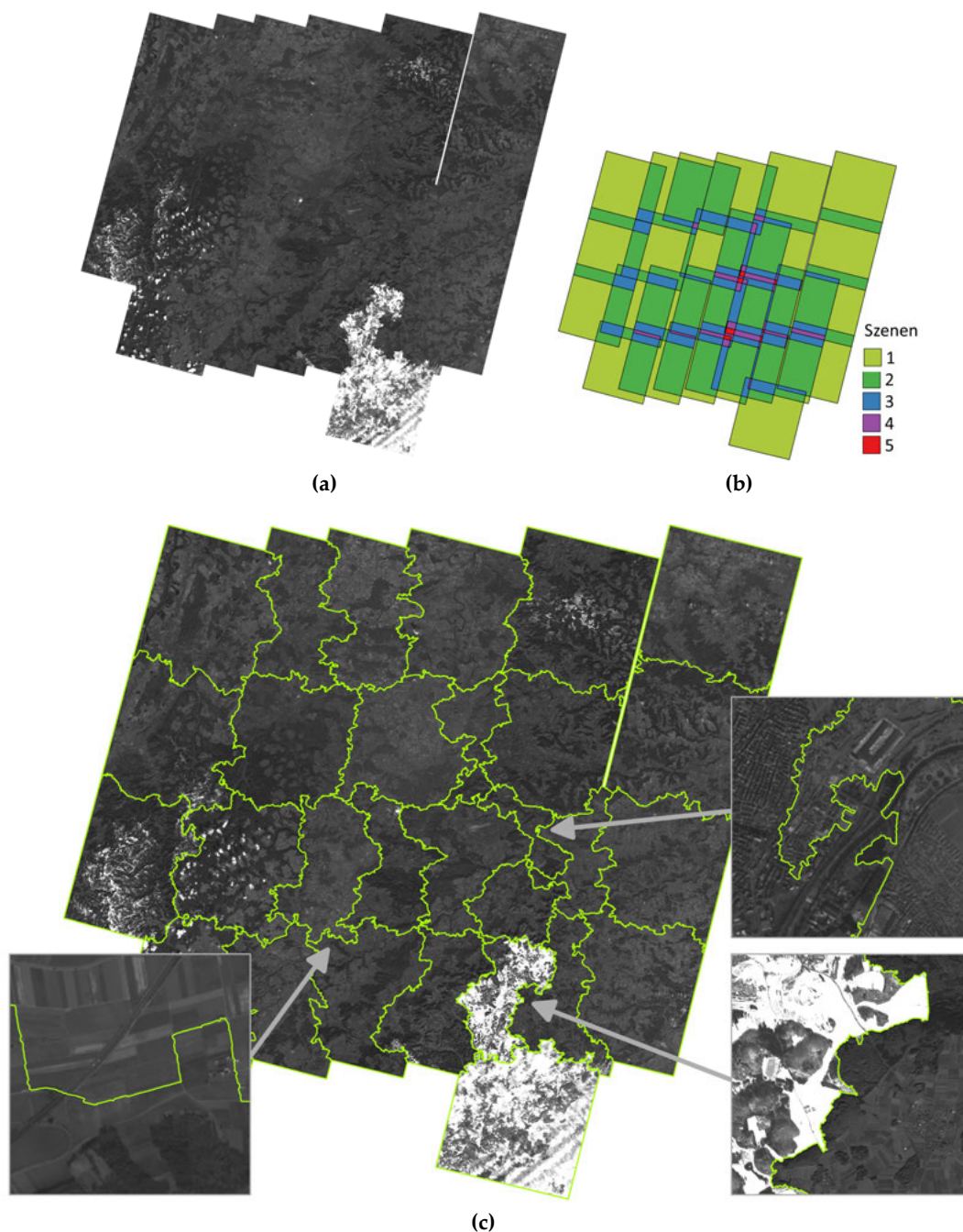
# Ergebnisse

---

*MSeam* wurde anhand verschiedener panchromatischer und multispektraler Testszenen evaluiert. Um die Qualität der gefunden Nahtlinien beurteilen zu können, ist es ausreichend, eine Auswahl weniger Szenen zu betrachten, was regelmäßig und begleitend zur Entwicklung von *MSeam* erfolgt ist. Um die Eignung von *MSeam* für die Berechnung ausgedehnter Mosaiken zu überprüfen, wurden aber auch größere Datensätze betrachtet. Es wurden hauptsächlich drei Testdatensätze genutzt: 1) Vier ALOS AVNIR-2 [28] Multispektralufnahmen der Umgebung von Innsbruck (Österreich), die in verschiedenen Jahreszeiten aufgenommen wurden, 2) 28 panchromatische Cartosat-1 [40] (IRS-P5) Fotos des Großraums Stuttgart und 3) verschiedene Resourcesat-1 [34] (IRS-P6 LISS III) Aufnahmen (multispektral, kein Blaukanal) aus dem Image-2006 Projekt [41] der ESA. Aufbauend auf dem letztgenannten Datensatz wurden ausgedehnte Mosaiken aus bis zu 291 Szenen berechnet. Dieser Abschnitt stellt eine Auswahl der betrachteten Testfälle vor. Dabei beschränkt sich die Betrachtung auf den Standardablauf für die vollautomatische Mosaikierung. Auf weitere Verfahren zur Bestimmung der Salienz von Bildmerkmalen und somit der Platzierung der Nahtlinien, wie sie im Rahmen der vielfältigen Konfigurationsoptionen *MSeams* zur Verfügung stehen, wird nicht näher eingegangen.

## 6.1 Hochaufgelöste panchromatische Aufnahmen

Für das Testen von *MSeam* mit hochaufgelösten panchromatischen Aufnahmen wurde ein Datensatz aus 28 Cartosat-1 Aufnahmen aus den Jahren 2008 bis 2012 ausgewählt, der die Umgebung von Stuttgart zeigt. Die geometrische Auflösung beträgt 2,5 m. Einige der Szenen zeigen großflächige Schneebedeckung oder viele Wolken, weshalb es in den Überlappungsbereichen teilweise zu deutlichen Unterschieden in den jeweiligen Inhalten kommt. Davon abgesehen, weisen die Szenen eine weitestgehend einheitliche Helligkeits- und Kontrastverteilung auf. Das finale Mosaik ist in Abbildung 6.1 abgebildet: Bildstrukturen in der Größenordnung weniger Pixel bereiten *MSeam* Schwierigkeiten. In diesem Fall verläuft die Nahtlinie gelegentlich nicht sauber entlang ausgedehnter, aber schwächer ausgeprägten Geländestrukturen. Ansonsten sind die Ergebnisse wie gewünscht.



**Abbildung 6.1:** Mosaik aus 28 panchromatischen Cartosat-1 [40] Aufnahmen, die von 2008 bis 2012 entstanden sind. Zu sehen sind Stuttgart und Umgebung. In (b) sind die Lage der Szenen und die zugehörigen Überlappungsbereiche farblich dargestellt. Die Helligkeits- und Kontrastverteilung ist weitestgehend homogen, weshalb die Nahtlinien ohne gesonderte Kennzeichnung nicht sichtbar sind (a). In (c) wurden die Nahtlinien hervorgehoben. In den drei Detaildarstellungen ist zu sehen, dass die Nahtlinie präzise sowohl den Schneefeldern und Waldrändern (unten rechts), als auch den Feldgrenzen beziehungsweise Feldwegen folgt (unten links). Schwer tut sich *MSeam* bei Vorhandensein einer Vielzahl feiner Strukturen (mitte rechts).



## 6.2 RGB-Mosaik

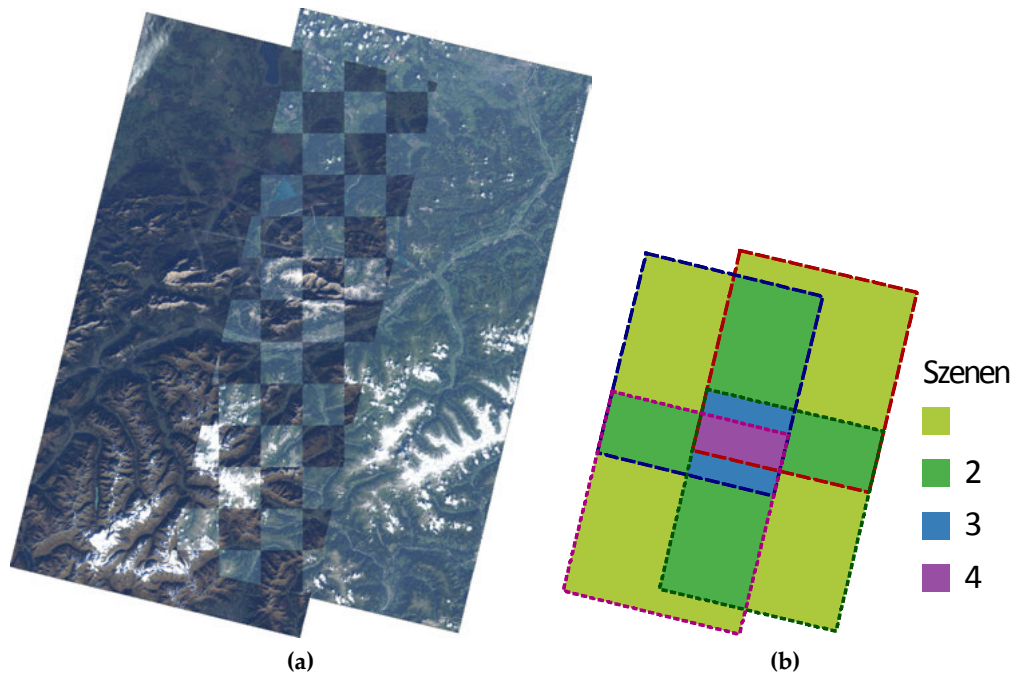
Viele Tests wurden mit einem Satz von vier ALOS AVNIR-2 [28] Aufnahmen durchgeführt (Gegend um Innsbruck, Österreich und Isarwinkel, Deutschland): Je zwei der Aufnahmen wurden zu unterschiedlichen Jahreszeiten aufgenommen (2006-10-27 und 2007-07-13), weshalb es insbesondere im Bereich der Berge deutliche Vegetationsunterschiede gibt. Auch führt die Isar unterschiedliche Wassermengen, was für ein uneinheitliches Erscheinungsbild sorgt. Nicht zuletzt ist die Beleuchtung sehr verschieden (deutlicher Schattenwurf in den linken Szenen). Um die Unterschiede zu illustrieren, wurden in Abbildung 6.2 die Szenen schachbrettartig übereinandergelegt. Je zwei der Szenen wurden dem selben Aufnahmestreifen entnommen, dann aber unabhängig orthorektifiziert. Die Genauigkeit der Orthorektifizierung ist üblicherweise sehr hoch, weshalb die Unterschiede von Szenen, die aus dem selben Aufnahmestreifen stammen, sehr gering sind. Es wäre somit denkbar, Überdeckungen von Szenen die dem selben Streifen entnommen wurden, anhand der Metadaten zu identifizieren, um dann die Szenen im Vorfeld der Mosaikierung durch einfaches Übereinanderlegen zu vereinigen.

Im Zuge des Standardablaufs für die vollautomatische Mosaikierung wird der Vorfilter nicht in Abhängigkeit von der Auflösung der jeweiligen Szenen bestimmt. Dies wurde damit begründet, dass der verfügbare Detailgrad bestmöglich ausgeschöpft werden soll (Abschnitt 5.4). Um dies zu zeigen, wurde das Mosaik neben den original aufgelösten Szenen (10 m geometrische Auflösung) auch für einen Datensatz berechnet, der auf 40 m geometrische Auflösung skaliert wurde. Die Ergebnisse werden in Abbildung 6.3 gegenüber gestellt. Auch wurde das Mosaik zusätzlich mit einer vorangegangenen radiometrischen Angleichung berechnet (Abbildung 6.4). In der selben Abbildung wird auch das Ergebnis eines optional möglichen *Blending* gezeigt: Problematisch sind diesbezüglich Nahtlinien, die in der Nähe von Bildbereichen verlaufen, für die nur eine Szene zur Verfügung steht. Ein weiträumiges Überblenden zwischen benachbarten Beiträgen ist dann nicht möglich, wodurch Artefakte begünstigt werden. Schwierigkeiten bereiten auch deutliche Unterschieden zwischen den Szenen, die durch eine Nahtlinie abgegrenzt werden.

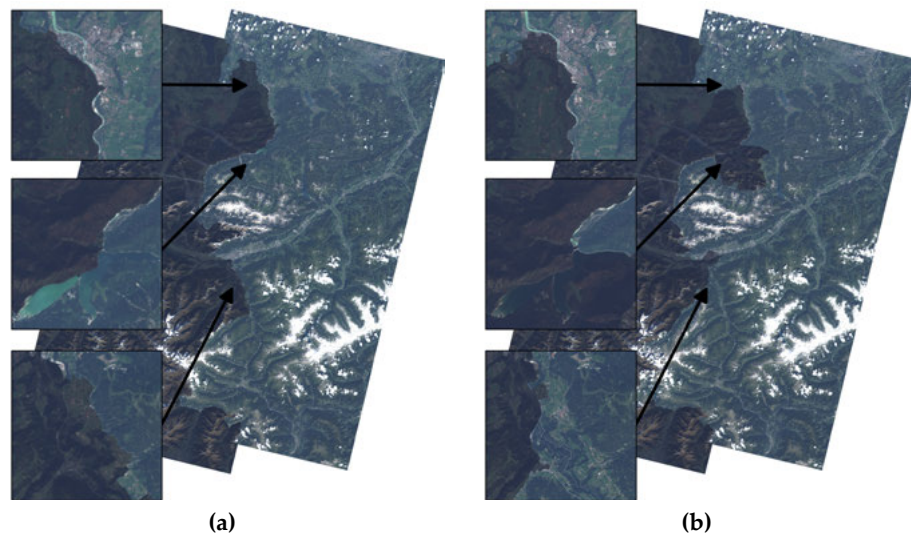
Bei der Mosaikierung der vier Aufnahmen wurden nur die Kanäle Rot, Grün und Blau berücksichtigt. Ein zusätzlich vorhandener Infrarotkanal wurde ignoriert. Prinzipiell ist es sinnvoll sich bei der Mosaikierung auf die für eine nachfolgende Analyse oder Visualisierung relevanten Kanäle zu beschränken. Andernfalls ergibt sich möglicherweise ein, für die Anwendung ungeeigneter Verlauf der Nahtlinien.

## 6.3 Image-2006 - Mosaikieren großer Datensätze

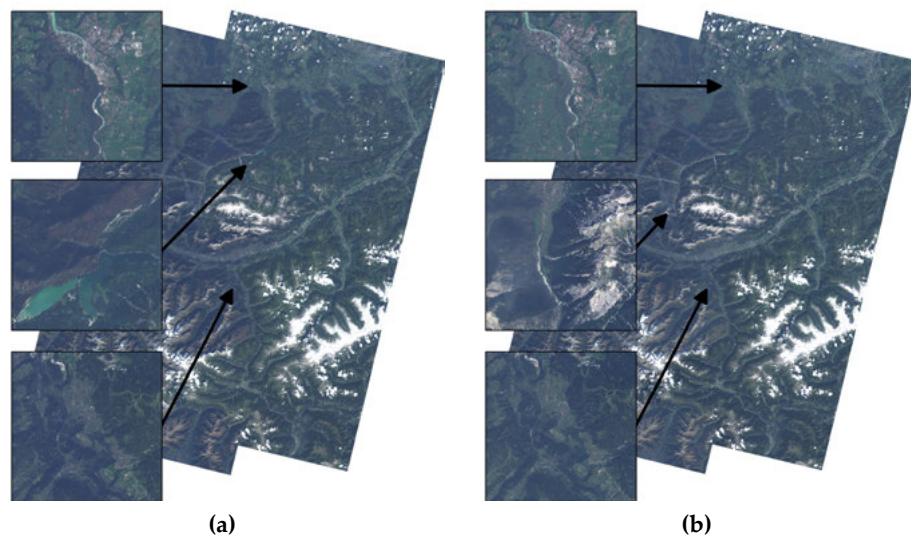
Der umfassendste, zur Verfügung stehende Datensatz entstammt dem Image-2006 [41] Projekt der ESA. Getestet wurde *MSeam* auf einer Auswahl von Resourcesat-1 [34] Szenen mit



**Abbildung 6.2:** Vier Szenen aus der Region Innsbruck, Österreich einschließlich Isarwinkel (Deutschland) wurden zum Testen ausgewählt. Es handelt sich um vier ALOS AVNIR-2 [28] Aufnahmen, die aus zwei Aufnahmestreifen entnommen wurden. Die Zeitpunkte, zu denen die Streifen aufgenommen wurden, liegen in verschiedenen Jahreszeiten (2006-10-27 und 2007-07-13), weshalb das Erscheinungsbilder der Szenen verschieden ist: Zum Beispiel ist die Vegetation in den Bergregionen unterschiedlich, auch führt die Isar (Fluss im oberen Drittel des Überlappungsbereichs) unterschiedliche Wassermengen. Hinzu kommt ein deutlicher Schattenwurf in den linken Szenen. Allgemein ist die Radiometrie sehr inhomogen. Für die Darstellung wurden die Szenen schachbrettartig übereinandergelegt (a), wobei die Aufnahmen zur besseren Visualisierung einheitlich aufgehellt wurden. Ein Quadrat des Schachbretts hat eine Kantenlänge von 1000 Pixeln. In (b) ist der Überdeckungsgrad der Szenen dargestellt.



**Abbildung 6.3:** Der Datensatz (Abbildung 6.2) wurde einmal in 40 m geometrische Auflösung (a) und ein weiteres Mal (b) in der Originalauflösung (10 m) mosaikiert. Ungeachtet teils deutlicher Bildunterschiede, findet *MSeam* jeweils geeignete Strukturen. Strukturen in der Größenordnung einzelner Pixel können das Ergebnis jedoch geringfügig beeinträchtigen (a: Ausschnitt links oben).



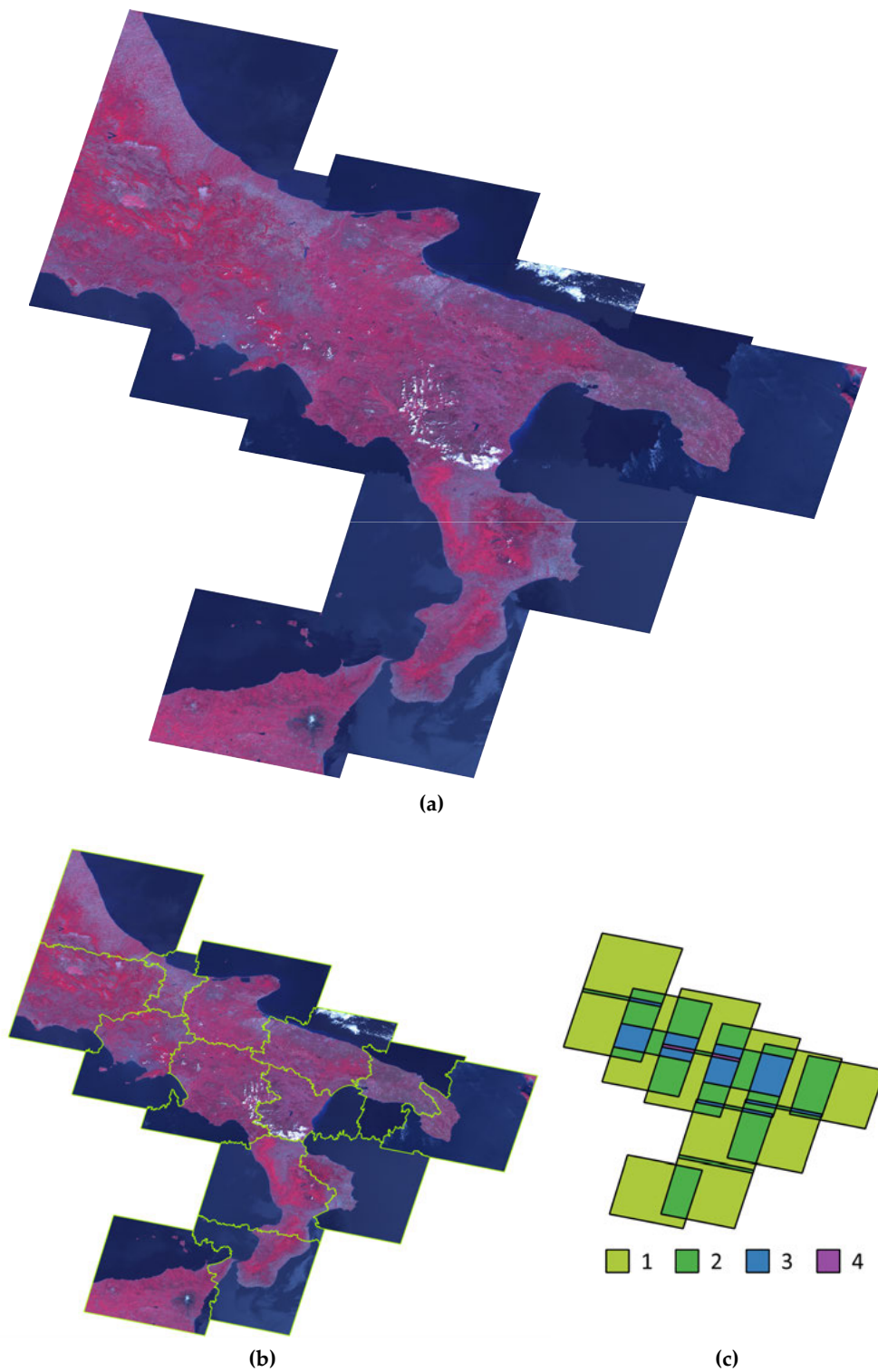
**Abbildung 6.4:** Der Datensatz (Abbildung 6.2, 40 m geometrische Auflösung) wurde vor der Bestimmung des Mosaiks radiometrisch angeglichen (a). Die Platzierung der Nahtlinien ändert sich geringfügig, bleibt aber weitestgehend unverändert. Wieder bereiten feine Strukturen mit scharfen Kanten Schwierigkeiten (a: oben links). Zusätzlich wurde ein *Blending* mit der *Multiblend*-Software [S6] durchgeführt (b). Verläuft die Seamline in der Nähe von Bildbereichen, für die nur eine Szene verfügbar ist, fällt das Ergebnis des *Blending* nur mittelmäßig aus (b: mittlerer Ausschnitt).

den Kanälen kurzwelliges Infrarot (SWIR), nahes Infrarot (NIR), Rot (R) und Grün (G) und einer geometrischen Auflösung von 25 m. Ausgedehnte Ausschnitte wurden ohne vorherige radiometrische Angleichung und unter Berücksichtigung aller vier Kanäle mosaikiert. Dementsprechend sind die Beiträge der einzelnen Szenen teilweise deutlich zu erkennen. Bei ausschnittsweiser Betrachtung, etwa als Kartenhintergrund, fallen die Übergänge jedoch kaum auf und stören nicht bei der Betrachtung, da die Nahtlinien weitestgehend Geländestrukturen folgen. Visuell besonders beeindruckend sind Mosaiken mit einheitlicher Radiometrie. Um dies zu demonstrieren, wurde eine Auswahl von 14 Orthofotos, die Kalabrien zeigen, mit *Xdibias* [57] radiometrisch angeglichen und unter Verwendung der Standardparameter mosaikiert. Das Ergebnis ist in Abbildung 6.5 dargestellt.

### 6.3.1 Mosaikierung großer Datensätze

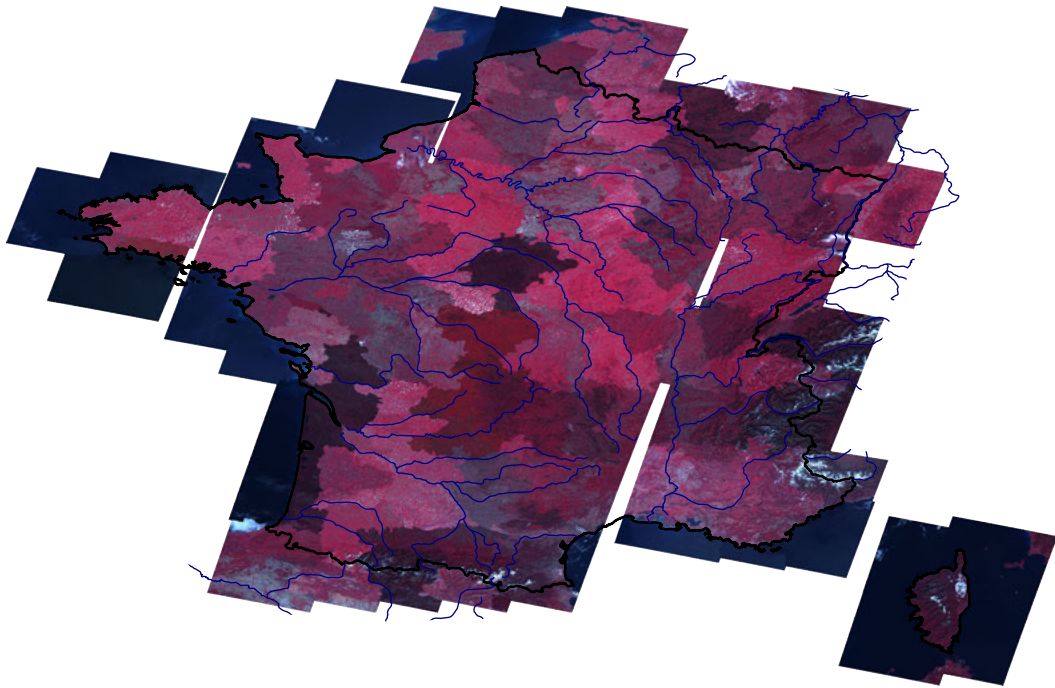
Auf Basis der Resourcesat-1 [34] Aufnahmen des Image-2006 Datensatzes [41] wurden zwei ausgedehnte Mosaiken erzeugt. Auf eine radiometrische Angleichung wurde verzichtet. Die Mosaikierung erfolgte unter Verwendung der universellen Standardparameter für die vollautomatische Mosaikierung. Ausgeführt wurde FAMOS dabei nicht unter *CentOS-5* [55], sondern unter *Arch-Linux* [51] und *Gentoo-Prefix* [512] (beides *Rolling Release* Distributionen) mit *Python-2.7* und der Möglichkeit die hochoptimierte *OpenCV-2* Bibliothek zu Nutzen. Beides verspricht eine tendenziell etwas schnellere Ausführung. Die Rechner waren jeweils mit 8 Gigabyte Hauptspeicher und einer Intel *Core-2* CPU ausgestattet.

Das erste Mosaik (Abbildung 6.6) besteht aus 109 Szenen mit vier Kanälen (Grün, Rot, nahes Infrarot und mittleres Infrarot) und deckt Frankreich mit Ausnahme einiger kleiner Bereiche vollständig ab. Der Datensatz umfasst etwa 75 Gigabyte Bilddaten und hat eine Größe von 20,1 Gigapixel (13,3 Gigapixel innerhalb des Definitionsbereichs). 5,1 Gigapixel tragen zum fertigen Mosaik bei. Die einzelnen Szenen überlappen teilweise fast vollständig (Abbildung 4.1), weshalb das Mosaik aufwändig zu berechnen ist - aufgrund des globalen Charakters der Wasserscheidentransformation wird jede Überdeckung vollständig segmentiert, auch wenn das Ergebnis nur in einem kleinen Ausschnitt beibehalten werden kann (Abschnitt 4 und 5): Die zu bearbeitenden Überdeckungen haben eine akkumulierte Größe von 55,6 Gigapixeln (gewichtet mit dem Grad der Überdeckung). Da tatsächlich die minimal umgebenden Rechtecke der Überlappungsbereiche betrachtet werden, gilt es sogar bis zu 111,0 Gigapixel zu verarbeiten. Diese akkumulierte Betrachtung berücksichtigt jedoch nicht, dass es bei der Erzeugung des Maskenbilds möglich ist, redundante Berechnungen durch geeignetes Caching zu vermeiden. Die sequentielle Berechnung dauerte 9,1 Stunden wovon 1,5 Stunden (16%) auf das Lesen und Schreiben von Bilddaten entfielen. 0,9 Stunden (10%) wurden für die Bestimmung der Randbedingungen benötigt; die finale Bestimmung der Beiträge einzelner Szenen einschließlich dem Filtern von Artefakten dauerte 1,4 Stunden (16%). Unter dem Strich



**Abbildung 6.5:** (a) Mosaik aus 14 radiometrisch angeglichenen Resourcesat-1 [34] Aufnahmen aus dem Image-2006 Projekt [41]. Mosaikiert wurden die Kanäle nahes Infrarot, Rot und Grün (Falschfarbendarstellung). (b) Dasselbe Mosaik, jedoch mit überlagerten Nahtlinien. (c) Auftretende Überdeckungen (Überdeckungsgrad farblich gekennzeichnet).





**Abbildung 6.6:** Mosaik Frankreichs aus 109 Resourcesat-1 [34] Aufnahmen des Image-2006 Datensatzes [41]. Die Aufnahmen entstanden in den Jahren 2005 bis 2007. Die Abbildung zeigt die Kanäle nahes Infrarot, Rot und Grün in Falschfarbendarstellung, bei der Mosaikierung wurde zusätzlich der vierte Kanal (kurzwelliges Infrarot) berücksichtigt. Auf eine radiometrische Angleichung der Szenen wurde verzichtet, die meisten der Szenengrenzen sind daher sichtbar. Isolierte Bereiche wurden beim Rendern des Mosaiks ausgefiltert (Abschnitt 5). Einige der Nahtlinien folgen den Flußläufen (große Flüsse in blau überlagert).

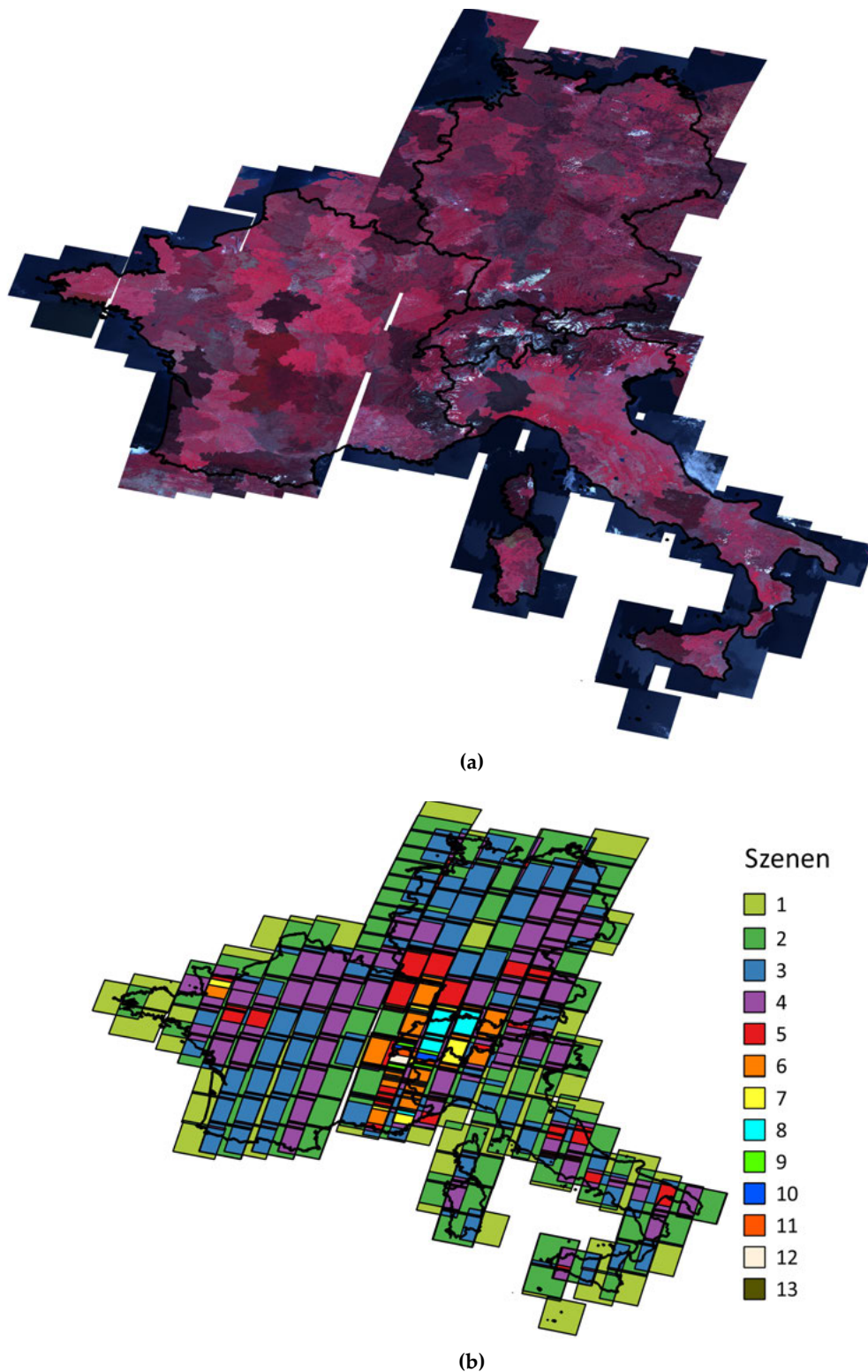
verbleiben demnach 5,3 Stunden (58%) für die Bestimmung und Segmentierung der Maskenbilder sowie für sonstige Operationen wie dem Lesen und dem Verwalten des *On-Disk-Cache*. Bezogen auf die minimal umgebenden Rechtecke der Überlappungsbereiche betrug der Durchsatz demnach etwas über 12 Gigapixel pro Stunde. Es wurde nicht ermittelt, wie viel Zeit die Indexierung der Szenen in Anspruch nahm. Das Mosaik wurde anschließend auf die Länder Deutschland, Schweiz und Italien ausgedehnt (Abbildung 6.7). Es wurden damit insgesamt 291 Szenen berücksichtigt, wobei 7 der Szenen keinen Beitrag zum finalen Mosaik leisten (siehe Abschnitt 5.2).

## 6.4 Laufzeit

Auf eine fundierte, theoretische Analyse der zu erwartenden Zeitkomplexität wurde verzichtet: Alle Pixeloperationen, die bei Verwendung von *MSeam* mit den universellen Standardparametern, genutzt werden, skalieren linear mit der Pixelanzahl. Diese Feststellung ist allerdings nicht ausreichend, um die Komplexität von *MSeam* abzuschätzen. Insbesondere ist nicht für alle verwendeten Vektoroperationen die Komplexität bekannt. Um dennoch ein Gefühl zu entwickeln, wie gut das Verfahren in der Praxis skaliert, wurde ein panchromatischer Datensatz bestehend aus 28 Szenen (circa 10 Gigabyte Bilddaten, 171 Überdeckungen, siehe auch Abschnitt 6.1) in verschiedenen Skalierungen mosaikiert und jeweils die normierte Laufzeit (Abbildung 6.8) bestimmt. Die Ausführung wurde dafür in jeder Skalenstufe fünffach wiederholt. Um den Einfluss des Betriebssystemcaches abschätzen zu können, wurde die Zeit, die das Lesen von Daten benötigte, ebenfalls aufgezeichnet. Das Ergebnis legt zusammen mit der Tatsache, dass nur Pixeloperationen linearer Komplexität genutzt werden, nahe, dass *MSeam* tatsächlich linear skaliert. Die relevanten Bezugsgrößen sind dabei die Anzahl der Konturpixel jeder Szene (Indexierung) und die Anzahl der Pixel in den Überdeckungen.

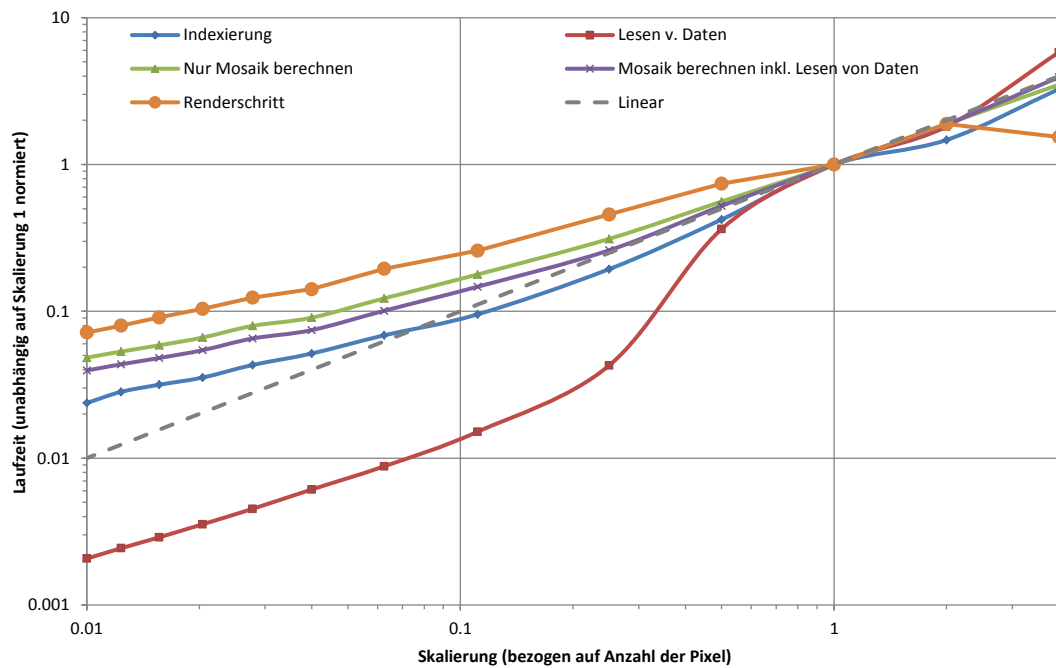
## 6.5 Weitere Verfahren

Der Schwerpunkt der Arbeit lag auf den Verfahren *Morphological Image Compositing* (Abschnitt 4) und dem weiterentwickelten *MSeam* (Abschnitt 5). Zusätzlich wurden zwei einfache Verfahren (Abschnitt 2) für das Zusammenfügen vieler Satellitenbilder zu einem Mosaik, sowie ein fundierter *Pixel Labeling* Ansatz (Abschnitt 3) betrachtet. Allen Methoden ist gemein, dass sie disjunkte Beiträge der einzelnen Szenen zum Mosaik bestimmen. Die beiden einfachen Verfahren eignen sich insbesondere für die Echtzeitvisualisierung. Ohne zusätzliches Überblenden benachbarter Szenen, fallen die Nahtlinien zwischen den Beiträgen jedoch deutlich auf. In Verbindung mit einem optionalen *Blending* lassen sich dennoch beeindruckende Ergebnisse erzielen.



**Abbildung 6.7:** Mosaik der Image-2006 (Resourcesat-1) Aufnahmen für die Länder Deutschland, Frankreich, Italien sowie Schweiz (a). Es wurden alle verfügbaren Kanäle bei der Berechnung des Mosaiks berücksichtigt. Der ausgewählte Datensatz umfasst 291 Szenen (b), sieben davon leisten allerdings keinen Beitrag zum Mosaik.





**Abbildung 6.8:** Ein Mosaik aus 28 panchromatischen Szenen wurde in verschiedenen Skalierungen mosaikiert und die Laufzeit bestimmt (gemittelt über je fünf Wiederholungen). Die Bilddaten der vollaufgelösten Szenen sind in Summe circa 10 Gigabyte groß. Die Berechnung erfolgte auf einer Workstation mit 8 Gigabyte Hauptspeicher. Bis einschließlich Skalierungsfaktor 1/4 ist die Datenmenge klein genug, sodass der Betriebssystemcache alle Szenen im Hauptspeicher vorhalten kann. Das erklärt den Sprung der *Lesen*-Kurve. Einer Anmerkung bedarf auch die Kurve für den Renderschritt: Da der Vorfilter nicht entsprechend der Skalierung der Bilddaten angepasst wurde, ist das resultierende Mosaik nicht von vornherein für alle Skalenstufen identisch. Dementsprechend kann sich ein unterschiedlich komplexer Verlauf der Nahtlinien ergeben. Da nun die Laufzeit des Renderschritt nicht von der Fläche der Beiträge, sondern von der Anzahl der Eckpunkte der betrachteten Polygone abhängt, ist das Abknicken der Laufzeitkurve nachvollziehbar.

*Pixel Labeling* beruht auf einer mathematischen Formulierung eines Optimierungsziels, entsprechend präzise lassen sich die Kriterien für die Wahl der Beiträge einzelner Szenen beschreiben. Die Laufzeitkomplexität des Verfahrens ist jedoch ungünstig um ausgedehnte Mosaiken zu berechnen. In dieser Hinsicht überzeugt *MSeam* mit (im Experiment) linearer Komplexität. Das angestrebte Ergebnis lässt sich dabei zwar weniger präzise beschreiben, aber die Ergebnisse sprechen insbesondere bei deutlichen Helligkeits- und Kontrastabweichungen für sich. Die guten Resultate lassen sich darauf zurückführen, dass sich die Abgrenzung benachbarter Beiträge zum Mosaik an Geländemerkmale orientiert. In Bereichen des Mosaiks, in denen sich viele Szenen überlappen, leidet jedoch unter Umständen die Genauigkeit, da der Suchraum für die *Seamlines* dann zu sehr eingeschränkt ist (siehe Abschnitt 4.3). Insofern wäre ein Vergleich mit *Pixel Labeling* unter Verwendung einer Kostenfunktion wie Formel 3.18 wünschenswert, mit der ähnliche Ergebnisse zu erwarten sind, wie sie *MSeam* anstrebt. Allerdings lässt sich eine solche Kostenfunktion nicht ohne weiteres mit den Anforderungen des verwendeten  $\alpha$ -Expansion-Algorithmus vereinbaren. In der Praxis wären dazu Anpassungen der, diesbezüglich genutzten Software-Bibliothek (keine Eigenentwicklung) notwendig geworden, die sich aus Zeitgründen jedoch nicht mehr realisieren ließen.

### Fazit

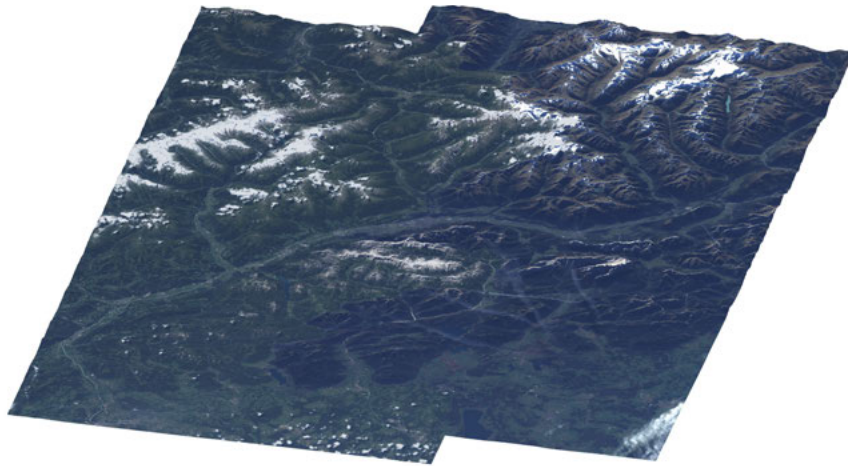
---

Mit *Morphological Image Compositing* und dem darauf aufbauenden *MSeam* steht ein Verfahren zur Verfügung, um ausgedehnte und hochaufgelöste Mosaiken aus optischen Satellitenbildern oder allgemeiner Orthofotos automatisch zu berechnen. Insbesondere setzen beide Ansätze keine einheitliche Helligkeits- und Kontrastverteilung voraus und stellen allgemein nur geringe Anforderungen an die Konsistenz sich überlappender Aufnahmen. Somit kommen beide Verfahren auch ohne eine Angleichung der Ausgangsbilder aus, weshalb auf eine *kosmetische* Vorverarbeitung der Bilddaten, zugunsten unverfälschter Radiometrie (Interpretation der Tonwerte) verzichtet werden kann. Entsprechend ist auch das Überblenden von Beiträgen benachbarter Szenen nicht notwendig; stattdessen werden disjunkte Beiträge ermittelt, die durch deutliche (saliente) Bildmerkmale voneinander abgegrenzt werden.

*MSeam* ergänzt das *Morphological Image Compositing* um die Möglichkeit, ein bestehendes Mosaik ohne Neuberechnung zu aktualisieren, also Einzelaufnahmen hinzuzufügen oder entfernen. Auch das Filtern möglicherweise auftretender Artefakte ist möglich. Die Salienz von Bildmerkmalen wird für multispektrale Aufnahmen fundiert bestimmt, wobei das Vorgehen bei Bedarf flexibel angepasst werden kann. Dazu wurde ein Prozess entworfen und beschrieben, der die strukturierte Ermittlung der Salienzmerkmale ermöglicht.

Auch bei inhomogenen Ausgangsbildern ergibt die von *MSeam* getroffene Auswahl der Bildbeiträge für den Betrachter einen schlüssigen Verlauf der Grenzen zwischen benachbarten Beiträgen. Beschränkt sich die Betrachtung auf Ausschnitte des zusammengesetzten Mosaiks, was zum Beispiel bei Verwendung der Bilddaten als Kartentextur in GIS-Systemen meistens der Fall ist, ergibt sich in jedem Fall, also auch bei abweichenden Bildinhalten, ein unauffälliger Übergang zwischen benachbarten Szenen. Bei Betrachtung des Gesamtmosaiks sticht hingegen eine, möglicherweise uneinheitliche, Helligkeits- und Kontrastverteilung ins Auge. Dies kann erwünscht sein, wenn es gilt, die physikalische Interpretation der Tonwerte nicht durch kosmetische Operationen zu verfälschen. Ist dies nicht notwendig, kann *MSeam* mit einer radiometrischen Angleichung der Ausgangsbilder kombiniert werden. Die Übergänge zwischen benachbarten Bildbeiträgen werden dann vollständig kaschiert. In jedem Fall ist *MSeam* mit und ohne eine solche Angleichung universell einsetzbar.

Die Fähigkeiten von *MSeam* wurden anhand ausgewählter Beispiele präsentiert. Zusätz-



**Abbildung 7.1:** *MSeam* im praktischen Einsatz: Schrägansicht der Gegend um Innsbruck, Österreich. Die Ansicht wurde unter Verwendung von *MSeam* und *Xdibias* [57] berechnet. Grundlage sind vier ALOS AVNIR-2 [28] Aufnahmen.

lich wurden auch Teile des Image-2006 Datensatzes der ESA bearbeitet, wodurch die Eignung für große Datensätze gezeigt wurde. Die entstandene Software wurde bereits am DLR eingesetzt (Abbildung 7.1).

Abschließend soll noch eine Idee zur Weiterentwicklung des Verfahrens aufgezeigt werden: Häufig verlaufen die von *MSeam* bestimmten Nahtlinien entlang von Strukturen wie Flüssen oder Straßen. Tatsächlich sind solche langgestreckten Geländemerkmale gut geeignet, um die Beiträge verschiedener Szenen voneinander abzugrenzen. Nun sind eben solche Merkmale aber auch gut kartiert und ändern sich nicht kurzfristig. Es liegt somit Nahe, bei der Weiterentwicklung von *MSeam*, entsprechende Kartendaten mit einzubeziehen<sup>1</sup>. Aufbauend auf der im Rahmen dieser Masterarbeit entstandenen Softwareumgebung *FAMOS* und deren *Python*-API, wäre dies direkt möglich.

---

<sup>1</sup>Diesbezüglich lohnt ein Blick in Abschnitt 1.3, Verwandte Arbeiten, worin auch auf eine Arbeit Bezug genommen wird, die diesen Ansatz verfolgt.

# Implementierung

---

### A.1 FAMOS - Ein kurzer Überblick

Im Rahmen der Masterarbeit ist die Software *FAMOS - Fully Automated Mosaicking Study* entstanden. Diese besteht aus einem erweiterbaren *Python* [S16]-Framework und einer darauf aufbauenden Konsolenanwendung. *FAMOS* bietet die notwendige Infrastruktur zum Testen verschiedener Verfahren zum Mosaikieren von Satellitenbildern (Orthofotos). Die Implementierung erfolgte in *Python* und *C++*. *Boost::Python* [S4] wurde genutzt um beide Technologien zu vereinen.

Aufbauend auf allgemeinen und erweiterbaren Komponenten zur Verwaltung des Mosaikierungszustands und einer einfachen *Python*-API werden spezifische Verfahren als *Strategie-Plugin* (*MosaickingStrategy*) implementiert. Konkret wurden drei solcher Strategien umgesetzt: 1) Einfaches Übereinanderlegen von Szenen, 2) *Pixel Labeling* für kleine Mosaiken und schwerpunktmäßig 3) *MSeam*. Letzteres ist auch für die Erzeugung einer Voronoisegmentierung geeignet.

Bei der Verwaltung des Mosaikierungszustands wird rege von Vektordaten Gebrauch gemacht, wobei die notwendigen Vektoroperationen sowohl für die Nutzung in *Python*, als auch unter *C++* verfügbar sind. Bei der Implementierung lag der Schwerpunkt auf der Handhabung von Pixelmasken, die als Polygon dargestellt werden. Darüberhinaus wurden einige spezielle Bildverarbeitungsroutinen implementiert und eine, in der Anwendung unkomplizierte, *Caching-API* geschaffen. Diese macht unter anderem von sogenannten *Python Decorators* Gebrauch, womit bestehende Funktions- oder Methodenaufrufe transparent um *Caching*-Funktionalität erweitert werden können. Desweiteren stehen noch etliche weitere Werkzeuge zur Verfügung, die nicht im Umfang einer *Python*-Installation enthalten sind.

Zur Verwaltung des Mosaikierungszustands wird die *In-Process*-Datenbank *SQLite* [S8] mit *SPATIALITE* [S2]-Erweiterung für den Umgang mit Geodaten genutzt. Aufgrund einer geschichteten Architektur und der Nutzung des ORM-Mappers und *SQL-Toolkits* *SQLAlchemy* [S14], ließe sich diese jedoch ohne viel Aufwand gegen ein mächtigeres Datenbanksystem austauschen.

Neben *SQLAlchemy* [S14] (und *GEOAlchemy* [S10] für die zusätzliche Unterstützung von Geodaten) wurde bei der Umsetzung noch auf eine handvoll weiterer Pakete beziehungs-

weise Bibliotheken zurückgegriffen. Dazu zählen insbesondere (die Aufzählung ist nicht abschließend) *NumPy* [S20] (effiziente *Array*-Implementierung), *SciPy* [S9] (wissenschaftliches Rechnen) und *GEOS* [S15] beziehungsweise *Shapely* [S19] (Handhabung, insbesondere zweidimensionaler Geometrien). Die Implementierung des *Pixel Labeling* greift verwendet die *GCO*-Bibliothek [S17].

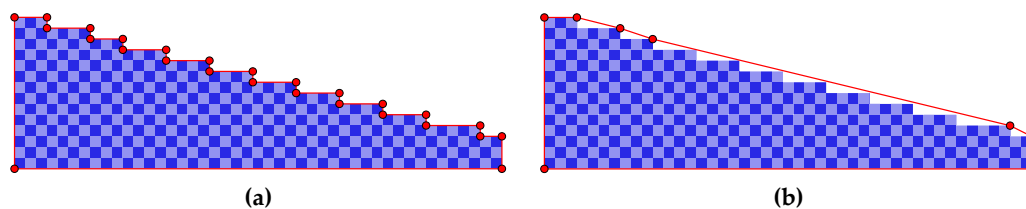
## A.2 Polymask - Pixelmasken effizient als Polygon darstellen

Pixelmasken, also eine Auswahl zusammengehörender Pixel (nicht zwangsläufig zusammenhängend), werden von *FAMOS* bevorzugt als Spezialfall eines Polygons betrachtet. Diese Sichtweise ermöglicht insbesondere die Nutzung ausgereifter Datenbanksysteme mit Unterstützung für Geodaten und die Verwendung effizienter Operationen aus dem Bereich der algorithmischen Geometrie. Insbesondere binäre morphologische Operationen lassen sich gut auf die Polygondarstellung übertragen. In jedem Fall ist die Speicherkomplexität deutlich geringer als bei Durchführung der Operation auf den entsprechenden Rasterbildern (Anzahl der benötigten Eckpunkte in der Größenordnung der Quadratwurzel der Pixelzahl).

Eine solche Polygondarstellung wird nachfolgend als *Polymask* bezeichnet. Dabei wird zwischen einer komprimierten und einer unkomprimierten (normierten) Darstellung unterschieden. In jedem Fall werden quadratische Pixel der Kantenlänge eins betrachtet. Die Koordinaten eines Punktes legen die Lage des zugehörigen Mittelpunkts fest. Die Ausdehnung eines Pixels mit den Koordinaten  $m$  (*Reihe*) und  $n$  (*Spalte*) entspricht somit den offenen Intervallen  $(m - 0,5; m + 0,5)$  entlang der vertikalen Achse und  $(n - 0,5; n + 0,5)$  entlang der horizontalen Achse. Ein Pixel ist Teil der Maske, wenn es vollständig innerhalb des zugehörigen Polygons liegt. Daraus ergibt sich auch die Unterscheidung zwischen komprimiert und normiert (Abbildung A.1): Letzteres erfordert, dass der Rand des zur *Polymask* gehörenden Polygons exakt dem Randverlauf der repräsentierten Pixelmaske entspricht. Ist dieses Kriterium nicht erfüllt, ist von einer komprimierten *Polymask* die Rede. Werden die zugehörigen Polygone betrachtet, entspricht die komprimierte Darstellung einer konservativen Approximierung der normierten Darstellung, mit der Bedingung, dass exakt die selben Pixel vollständig in den jeweiligen Polygonen enthalten sind. Unter dieser Voraussetzung existiert für jede komprimierte *Polymask* eine eindeutig bestimmte, normierte Darstellung. Die Benennung ist darauf zurückzuführen, dass die Approximation prinzipiell mit weniger Eckpunkten auskommt.

Beliebige Polygone lassen sich als komprimierte *Polymask* auffassen. Die Dekomprimierung wurde in *FAMOS* als *Sweep*-Verfahren<sup>1</sup> umgesetzt, mit dem die normierte Darstellung zeilenweise aufgebaut wird. *FAMOS* enthält auch einen Algorithmus um

<sup>1</sup>Zeilenweises „Abtasten“ eines Polygons entlang einer ausgezeichneten Achse



**Abbildung A.1:** Polymask-Darstellung einer Pixelmaske. Je ein Quadrat entspricht einem Pixel der Maske. In (a) ist die normierte Polymask-Darstellung zu sehen, (b) zeigt eine mögliche komprimierte Polymask-Darstellung. Hierbei wurde der implementierte Kompressionsalgorithmus auf (a) angewandt. Das Ergebnis ist eine konservative Approximation des in (a) gezeigten Polygons. Wegen der besonderen Eigenschaften einer Polymask, kann daraus auch die normierte Darstellung wieder eindeutig abgeleitet werden.

die normierte Darstellung durch geeignetes Weglassen von Eckpunkten in eine komprimierte *Polymask* zu überführen.

### A.3 Wasserscheidentransformation

In Abschnitt 5.3.1 wurde die markergesteuerte Wasserscheidentransformation erläutert. Die Überdeckungen werden auf Basis dieses Verfahrens segmentiert, weshalb die gewählte Implementierung entscheidenden Einfluss auf das finale Mosaik hat. Tatsächlich existieren viele Möglichkeiten, die Wasserscheidentransformation umzusetzen. Bevorzugt sollte ein Verfahren genutzt werden, welches das in Abschnitt 5.3.1 geschilderte Prinzip möglichst genau umsetzt. Das von *MSeam* genutzte Verfahren ist austauschbar und wird optional in der XML-Konfiguration festgelegt. Standardmäßig wird eine Implementierung genutzt, die auf dem Quellcode der *OpenCV*-Bibliothek aufbaut [S13].

Die *OpenCV*-Implementierung entspricht prinzipiell den geschilderten Anforderungen *MSeams*, ist allerdings auch für das Mosaikieren mehrkanaliger Bilder geeignet, weshalb die Pixel als Vektoren interpretiert werden. Da für Vektoren keine Ordnungsrelation definiert ist, fällt es schwer, ein solches Mehrkanalbild als Relief zu interpretieren. Entsprechend kann die Reihenfolge, in der die Keimzellen auf angrenzende Pixel ausgedehnt werden, nicht anhand der Pixelwerte bestimmt werden (für einkanalige Bilder entsprechen die Pixelwerte der jeweiligen Höhe des gedachten Reliefs). Die Distanz je zweier Pixel lässt sich allerdings auch in der vektoriellen Betrachtung eindeutig bestimmen. Alle Pixel, die an eine der Keimzellen angrenzen, werden nun entsprechend der Distanz zu den direkt benachbarten Pixeln (4-Nachbarschaft), die bereits einer Keimzelle zugeordnet wurden, in einer Prioritätswarteschlange verwaltet. Diese legt fest, in welcher

Reihenfolge und Richtung die Keimzellen ausgedehnt werden. Dieses Vorgehen liefert nicht unter allen Umständen das gewünschte Ergebnis: Je nach anfänglicher Wahl der Keimzellen und der Struktur des zu segmentierenden Bildes ergibt sich ein Ergebnis, das von der Betrachtung in Abschnitt 5.3.1 abweicht. Für Grauwertbilder, die sich als Relief auffassen lassen, bedeutet das, dass die, am Ende auf den Bereich des gesamten Bildes ausgedehnten, Keimzellen, nicht notwendigerweise durch die Wasserscheiden des betrachteten Reliefs abgegrenzt werden.

Da nun *MSeam* ausschließlich einkanalige Maskenbilder nutzt, wurde der *OpenCV*-Quellcode dahingehend angepasst, dass die Priorisierung anhand der Grauwerte der Pixel erfolgt. Die Priorität eines Pixels bestimmt sich dabei als Maximum des zugehörigen Grauwerts und der Priorität des zuletzt gekennzeichneten Pixels. Dementsprechend wird ein steigender *Grundwasserspiegel* simuliert, der an den anfänglich gekennzeichneten Keimzellen an die Oberfläche tritt. Ausgehend von diesen markierten Bildpunkten wird die Umgebung geflutet. Dieses Vorgehen ignoriert die Struktur von Senken, in die die Flut *hinüberschwappt*, deren Tiefpunkt also unterhalb des aktuellen Wasserspiegels liegt und die bisher noch nicht von der Flut erreicht wurden: Ausgehend von den Rändern einer solchen Senke, breitet sich die Flut gleichberechtigt, also mit gleicher Geschwindigkeit, aus. Werden dabei unterschiedlich markierte Keimzellen in den Bereich der Senke propagiert, entsteht somit eine Voronoi-Aufteilung, die sich nicht an der Struktur des Reliefs orientiert. Die umgesetzte Implementierung kann die Struktur im Bereich einer solchen Senke bei der Bestimmung der Prioritätswerte (optional) berücksichtigen.

Die skizzierte Variante der WST kennzeichnet die Wasserscheiden explizit, diese müssen also nachträglich einer der Keimzellen zugeordnet werden, um eine vollständige Aufteilung des Bilds (im Kontext von *MSeam* eine Überdeckung) zu erhalten. Dies erfolgt entweder willkürlich nach dem numerischen Wert des zugehörigen *Markers* oder anhand von Bildmerkmalen (des Orthofotos): Zum Beispiel kann die Zuordnung so gewählt werden, dass die Tonwertdifferenz entlang der resultierenden Nahtlinie minimiert wird.



# Hinweise zur Benutzung

---

Dem Nutzer präsentiert sich *FAMOS* (Fully Automated Mosaicking Study) als Kommandozeilenanwendung, wobei Projektparameter in einer *XML*-Steuerdatei spezifiziert werden. Auf die Kommandozeilenargumente soll in diesem Abschnitt nicht näher eingegangen werden, diesbezüglich sei auf die integrierte Hilfefunktion (Parameter `--help`) verwiesen. Wegen weitergehender Informationen zur *XML*-Projektdatei wird ebenfalls auf die in jeder *FAMOS*-Installation enthaltene Datei `example.xml` sowie das zugehörige *XML*-Schema `XmlMosaicDescription.xsd` verwiesen. Schwerpunkt dieses Abschnitts sind Hinweise zur Benutzung der *MSeam*-Strategie (siehe auch Abschnitt 5). *MSeam* dehnt Bildbereiche, die zu Beginn bereits eindeutig einem Ursprungsbild zugeordnet werden können, sukzessive auf angrenzende Regionen aus, in denen sich mehrere Bilder überdecken. Das durchgeführte Region Growing wird dabei durch ein Maskenbild gesteuert. Der Prozess (Abschnitt 5.3.2), in dem eben dieses erzeugt wird, ist frei konfigurierbar: Dazu können verschiedene Filter und Operationen kombiniert werden, die in diesem Abschnitt kurz aufgezählt und erläutert werden. Auf die *Python-API*, die das Implementieren eigener Filter ermöglicht, wird nicht eingegangen.

Zu mosaikierende Orthofotos müssen im *Xdibias*-Datenformat vorliegen, ein am DLR entwickeltes, proprietäres Bildformat. Damit ein Datensatz sinnvoll mit *MSeam* verarbeitet werden kann, muss für jede Szene gelten, dass sie nicht vollständig von einer weiteren Szenen des Datensatzes überdeckt wird. Eine vollständige Überdeckung durch mehrere Szenen zusammen, ist hingegen kein Problem (Abschnitt 5.2).

## B.1 XML-Projektbeschreibung

Die Steuerdatei gliedert sich in drei Abschnitte, `head`, `metadata` und `data`. Im ersten Abschnitt, `head`, erfolgt die Festlegung aller wesentlichen Projektparameter. Der Abschnitt `metadata` erlaubt die Festlegung erlaubter Bildparameter (etwa Auflösung) und die Auswahl der, bei der Mosaikierung zu berücksichtigenden, Kanäle. Diese Angaben sind optional, da alle Informationen gegebenenfalls aus einer der Szenen abgeleitet werden können. Der Knoten `data` dient der Auflistung des Datensatzes. Nachfolgend wird auf die Knoten `metadata` und `data` nicht weiter eingegangen, diesbezüglich sei auf das Beispielpjekt verwiesen. Die Betrachtung in diesem Abschnitt beschränkt sich im wesentlichen auf die Konfiguration der *MSeam*-Strategie (`strategyArgs`-Knoten).

```

1 <mosaicDescription xmlns="ns:episource.de/schema/2013/mosaic">
2   <head>
3     <!-- Projektverzeichnis -->
4     <workDirectory>${XMLDIR}</workDirectory>
5     <!-- MSeam-Strategie verwenden -->
6     <strategy>
7       famos.strategy.MorphologicalSeamMosaickingStrategy
8     </strategy>
9     <!-- Jede Strategie definiert Parameter, die angepasst werden
10        können. Eine Strategie muss für jeden Parameter Standard-
11        werte vorsehen, die Angabe von Parametern ist somit
12        optional. -->
13     <!-- Die Angabe der einzelnen Argumente erfolgt in Python-
14        Syntax; beliebige Python-Konstrukte sind erlaubt. Dies
15        hat allerdings zur Folge, dass "True" und "False" anstatt
16        der XML-typischen kleinen Schreibweise "true" ("false")
17        verwendet werden müssen. -->
18     <strategyArgs>
19       <!-- Analog zu Python import-Ausdrücken; "from" wird auch
20        unterstützt -->
21       <import as="mseam">famos.strategy.mseam</import>
22       <!-- Wesentliche Parameter: prefilter, mask_algorithm,
23        mask_filter, pixel_difference_algorithm, filter_holes,
24        chunk_size segmentation_algorithm; Beispiel: -->
25       <arg key="prefilter">mseam.GaussianFilter(ksize=5)</arg>
26     </strategyArgs>
27     <!-- Standardmäßig erfolgt keine Ausgabe, es wird lediglich
28        die Zustandsdatenbank aktualisiert -->
29     <outputFormatter>famos.out.MultiFormatter</outputFormatter>
30     <outputFormatterArgs>
31       <!-- Struktur analog zu strategyArgs -->
32       <import as="out">famos.out</import>
33       <arg key="formatter_cls_list">
34         [out.ImageFormatter, out.GeojsonFormatter]
35       </arg>
36     </outputFormatterArgs>
37     <!-- Optionale Angaben, etwa zu Caching, Datenbank, ... -->
38   </head>
39   <metadata>
40     <!-- Nur die gewählten Kanäle berücksichtigen -->
41     <channels>3 2 1</channels>
42     <!-- hier könnten weitere Bildparameter festgelegt werden -->
43   </metadata>
44   <data>
45     <xdibiasImage>${HOME}/.xdibias/local.area/img1</xdibiasImage>
46     <xdibiasImage>${XMLDIR}/ganz/viele/Szenen/img*</xdibiasImage>
47     <!-- weitere Szenen -->
48   </data>
49 </mosaicDescription>

```

Listing B.1: XML-Projektdatei (Auszug)

Die Konfiguration einer Strategie erfolgt mittels des `strategyArgs` Knoten. Dieser ermöglicht die Verwendung von beliebigen Python-Ausdrücken. Analog zu gewöhnlichen Python-Skripten müssen Module, die verwendet werden sollen, importiert werden. Dies geschieht analog zum Python `import`-Ausdruck mit dem `import`-Knoten. Einzelne (benannte) Argumente werden mittels `arg`-Knoten festgelegt (siehe Listing B.1). Nachfolgend werden die wesentlichen Parameter der *MSeam*-Strategie aufgelistet und knapp beschrieben. Die Namen in Klammern beziehen sich auf Abschnitt 5.3.2; der Argumenttyp ist in Kursivschrift angegeben (Erläuterung der Filter in Teil B.2).

**prefilter** (*Vorfilter - Filter oder None*) Bei der Berechnung des Maskenbilds verwendeter Vorfilter. Standard: `GaussianFilter(ksize=5)`

**mask\_algorithm** (*Maskenalgorithmus - Maskenalgorithmus*) Bestimmung eines Salienzmerkmals. Standard: `ScharrGradientMaskAlgorithm(multi_channel_mode=MULTICHANNEL_GRADIENT_MODE.DIZENZO)`

**mask\_filter** (*Maskenfilter - Filter oder None*) Geeignete Berücksichtigung von geometrischen und inhaltlichen Unterschieden sich überlappender Szenen bei der Bestimmung des Maskenbilds. Standard: `GaussianFilter(ksize=7)`

**pixel\_difference\_algorithm** (*Postfilter - Postfilter oder None*) Skalierung des Maskenbilds anhand von Kriterien, die alle zu einer Überdeckung beitragenden Szenen gemeinsam berücksichtigen (z.B. Tonwertunterschiede). Standard: `None`

**filter\_holes** (*Boolean*) Filtern von Artefakten im Renderschritt (Abschnitt 5.6). Standard: `True`

**chunk\_size** (*Integer*) Unter gewissen Voraussetzungen kann die Berechnung des Maskenbild in mehreren kleineren Ausschnitten erfolgen um die Nutzung des Hauptspeichers zu reduzieren. Die Ausschnitte sind quadratisch, wobei *chunk\_size* die Kantenlänge angibt. `0` deaktiviert dieses Verhalten. Standard: `5000`

**segmentation\_algorithm** (*Segmentierungsverfahren*) Zu verwendende Implementierung der markergesteuerten Wasserscheidentransformation. Standard: `MeyersWatershed(flooding=True, seam_objective=None, no_fill=False)`

## B.2 Implementierte Filter und Algorithmen

Der Prozess, in dem das gemeinsame Maskenbild für eine Überdeckung berechnet wird, gliedert sich in mehrere Teilabläufe, die vielfältig konfiguriert werden können. Dazu wurden bereits einige Filter und Algorithmen implementiert, die, entsprechend der jeweiligen Bedürfnisse, kombiniert werden können. Es gibt drei Klassen von Filtern beziehungsweise Algorithmen:

**1) Filter** Ein *Filter* bildet ein Eingabebild, auf nicht näher festgelegte Weise, auf ein transformiertes Ausgabebild ab. Die Anzahl der Kanäle und die Größe des Eingabebilds sind festgelegt. Die Größe des Ausgabebilds ist abhängig von der Konfiguration des Filters.

bilds dürfen sich dabei nicht ändern. Optional hat eine Filterimplementierung auch Zugriff auf die Ausgangsdaten des jeweiligen Zweigs des Maskenprozesses (siehe Abbildung 5.4). Ein Filter, der von dieser Möglichkeit Gebrauch macht, wird auch als *Maskenfilter* bezeichnet. Ein solcher *Maskenfilter* kann genutzt werden, um das Eingabild (dabei handelt es sich dann meistens um ein *vorläufiges*<sup>1</sup> Maskenbild) in Abhängigkeit von den Ausgangsdaten zu skalieren (die Grauwerte). Somit können etwa langgestreckte Kantenzüge besonders gewichtet werden.

**2) Maskenalgorithmus** Berechnet für jeden Bildpunkt des (vorgefilterten) Eingabebilds einen Salienzwert. Das Ergebnis ist ein einkanaliges, *vorläufiges*<sup>1</sup> Maskenbild.

**3) Nachfilter**<sup>2</sup> Letztlich ein spezieller *Filter*, der jedoch Zugriff auf *alle* Ausgangsdaten des Maskenprozesses hat (Abbildung 5.4).

Eine Auswahl möglicher Filter und Algorithmen wurde implementiert und kann genutzt werden, um die einzelnen Filterschritte, die bei der Berechnung des Maskenbilds durchlaufen werden, anzupassen. Eine Besonderheit stellt die *FilterChain* dar, die es erlaubt, innerhalb eines Filterschritts beliebige Filter zu kombinieren. Die folgenden Abschnitte geben einen Überblick über die verfügbaren Filter, wobei jeder nur kurz umrissen wird. Auf eine ausführliche Beschreibung der Parameter wird zugunsten der Übersichtlichkeit verzichtet.

### B.2.1 Filter

**GaussianFilter** Gauß'scher Weichzeichner beziehungsweise Tiefpassfilter. Es können entweder die Größe des Filterkernels oder die Standardabweichung der zugrundeliegenden Gaußkurve angegeben werden. Der jeweils fehlende Parameter wird geeignet bestimmt. Im Fall mehrkanaliger Eingabebilder, wird der Filter unabhängig auf jeden der Kanäle angewandt.

**MaxFilter** Bestimmt für jedes Pixel des Eingabebilds, den maximalen Tonwert innerhalb einer festzulegenden Umgebung und legt den Pixelwert des Ausgabebilds entsprechend fest. Jeder Kanal wird unabhängig gefiltert. Im Fall mehrkanaliger Eingabebilder, wird der Filter unabhängig auf jeden der Kanäle angewandt.

**CenteredMaxFilter** Wie *MaxFilter*, jedoch wird von jedem Pixelwert innerhalb der betrachteten Umgebung vor der Berechnung des Maximums zunächst die Manhattan-Distanz zum Zentrumspixel abgezogen.

**BilateralFilter** Nichtlinearer Filter, der Bilder weichzeichnet, dabei aber Objektkanten erhält. Es wird auf die *OpenCV*-Implementierung [S13] zurückgegriffen; in der zugehörigen Dokumentation findet sich eine ausführliche Erläuterung des Filters.

<sup>1</sup>Das Ergebnis des *Maskenalgorithmus* wird als *vorläufiges* Maskenbild bezeichnet um die Unterscheidung vom (*gemeinsamen*) Maskenbild zu ermöglichen, durch welches letztlich das Region Growing gesteuert wird - siehe Abschnitt 5.3.2.

<sup>2</sup>Im Quelltext und in der XML-Konfigurationsdatei als `pixel_difference_algorithm` bezeichnet

**KmeansClusteringFilter** Vereinfacht das Eingabebild, indem die Zahl auftretender Pixelwerte reduziert wird. Dazu wird eine festzulegende Anzahl an Clustern mit dem *K-Means*-Algorithmus bestimmt.

**QuasiFlatzoneFilter** Vereinfacht das Bild wie folgt: Zunächst werden Gruppen ähnlicher Pixel, die eine festgelegte Mindestgröße überschreiten, identifiziert. Im weiteren Verlauf werden die Gruppen unter Berücksichtigung eines Ähnlichkeitsmaßes um angrenzende Pixel erweitert, bis keine weiteren Bildpunkte verbleiben. Allen Pixeln einer Gruppe kann dann schließlich ein einheitlicher Pixelwert zugewiesen werden, zum Beispiel der durchschnittliche Tonwert dieser Gruppe. Beruht auf einem Vorschlag von Soille [22], [42].

## B.2.2 Maskenfilter

Jeder *Filter* kann auch als Maskenfilter eingesetzt werden. Darüberhinaus wurden zwei spezielle *Maskenfilter* implementiert:

**WeightedCannyMaskFilter** Verwendet intern das Ergebnis des *WeightedCannyMaskAlgorithm* (B.2.3) um Bereiche ausgedehnter Kantenzüge zu gewichten.

**LineRegionDetectorFilter** Nutzt ein Zwischenergebnis des sogenannten *Line Segment Detectors* [39], um ausgedehnte Kantenzüge zu identifizieren, deren Umgebung im zu filternden (*vorläufigen*) Maskenbild dann durch geeignetes Skalieren der Grauwerte gewichtet wird.

## B.2.3 Maskenalgorithmus

**ScharrGradientMaskAlgorithm** Aufbauend auf dem sogenannten *Scharr*-Operator, wird die Salienz als Rate der maximalen Bildänderung berechnet. Die Idee wird in Abschnitt 5.3.2 näher erläutert.

**SoilleMaskAlgorithm** Bestimmt die Salienz mit dem morphologischen Gradienten (Formel 4.9) beziehungsweise dessen Verallgemeinerung für Mehrkanalbilder (Formel 4.10).

**WeightedCannyMaskAlgorithm** Bestimmt Kantenzüge mit dem *Canny*-Kantendetektor [37], wobei die notwendigen Hysterese-Parameter in Abhängigkeit von Bildeigenschaften automatisch ermittelt werden. Die Salienz gefundener Kanten wird entsprechend der Fläche der zugehörigen *Kantenmarker* und einem Kompaktheitskriterium bestimmt (langgestreckte Kanten mit geringer Kompaktheit sollen bevorzugt werden). Damit dieser *Maskenalgorithmus* sinnvolle Ergebnisse liefern kann, muss eine Überdeckung jeweils vollständig und nicht in kleinen Ausschnitten verarbeitet werden (Option `chunk_size=0`; siehe Appendix B.1). Ein Beispiel eines

Maskenbilds, das mit diesem Algorithmus berechnet wurde, ist in Abbildung 5.9 dargestellt.

#### B.2.4 Nachfilter

Es wurde lediglich ein Nachfilter implementiert:

**LinearGWdiffFilter** skaliert das *gemeinsame* Maskenbild (die Grauwerte) unter Berücksichtigung der (für jedes Pixel unabhängig bestimmten) maximalen Tonwertabweichung zwischen den, an einer Überdeckung beteiligten Szenen. Dabei wird für jedes Pixel eine festzulegende Nachbarschaft betrachtet und darin die maximal auftretende Tonwertdifferenz bestimmt. Der Skalierungsfaktor bestimmt sich dann in Abhängigkeit vom Verhältnis der Tonwertdifferenz des jeweils betrachteten Pixels zu eben dieser maximalen Tonwertdifferenz und zwar dergestalt, dass der Skalierungsfaktor eins ist, wenn keine Tonwertabweichung festgestellt wurde; andernfalls ist er kleiner eins.

---

# Abbildungsverzeichnis

---

1.1	Definitionsbereich einer Szene . . . . .	8
2.1	Voronoi-segmentierung . . . . .	19
3.1	Ungerichteter Graph mit möglichem Schnitt . . . . .	23
3.2	<i>Pixel Labeling</i> : Ähnliche Tonwerte . . . . .	28
3.3	Zerlegung des Mosaiks in Zellen für das iterative Mosaikieren . . . . .	30
3.4	Vorsegmentierung . . . . .	32
4.1	Image-2006 Abdeckung Frankreichs (Resourcesat-1) . . . . .	36
4.2	Ablauf des <i>Morphological Image Compositing</i> . . . . .	39
4.3	<i>Morphological Image Compositing</i> illustriert . . . . .	42
4.4	<i>Morphological Image Compositing</i> - Vor- und Nachteile anhand der Ergebnisse von Soille . . . . .	45
5.1	Funktionsweise von <i>MSeam</i> . . . . .	50
5.2	Vollständig überdeckte Szenen können entfallen . . . . .	51
5.3	Iteratives Mosaikieren mit <i>MSeam</i> . . . . .	52
5.4	Bestimmung des Maskenbilds . . . . .	53
5.5	Bildsegmentierung mit der Wasserscheidentransformation . . . . .	55
5.6	Markergesteuerte Wasserscheidentransformation: Abgrenzung der Basins folgt teilweise nicht den „Wasserscheiden“ . . . . .	56
5.7	Scharr-Gradientenoperator . . . . .	60
5.8	Maskenbild (Scharr + Di Zenzo) . . . . .	61
5.9	Maskenbild (Canny) . . . . .	61
5.10	Entstehung von Artefakten . . . . .	66
6.1	Hochaufgelöstes panchromatisches Mosaik (Cartosat-1) . . . . .	70
6.2	Testgebiet: Isarwinkel (ALOS AVNIR-2) . . . . .	72
6.3	Echtfarben-Mosaik (ALOS AVNIR-2) . . . . .	73
6.4	Echtfarben-Mosaik (ALOS AVNIR-2, radiometrisch angeglichen) . . . . .	73
6.5	Image-2006 Mosaik (Kalabrien, radiometrisch angeglichen) . . . . .	75
6.6	Image-2006 Mosaik (Frankreich) . . . . .	76
6.7	Image-2006 Mosaik (Deutschland, Frankreich, Italien, Schweiz) . . . . .	78
6.8	Experimentelle Bestimmung der Zeitkomplexität . . . . .	79
7.1	<i>MSeam</i> im Einsatz: Schrägansicht der Gegend um Innsbruck . . . . .	82
A.1	Polymask . . . . .	85





---

## Software & Bibliotheken

---

Übersicht der Software und Programmbibliotheken sowie Frameworks, die in dieser Arbeit erwähnt werden.

- [S1] Aaron Griffin (2013)  
Arch Linux, *Rolling Release*  
*Linux-Distribution (Rolling Release Distribution), u. a. GPL*  
<http://www.archlinux.org/>
- [S2] Alessandro Furieri (2012)  
libspatialite 3.0.1  
*Ergänzt SQLite um Unterstützung für Geodaten, MPL tri-license*  
<http://www.gaia-gis.it/fossil/libspatialite/index>
- [S3] Atlantis Scientific Inc (2004)  
OpenEv 1.8  
*Softwarbibliothek und darauf aufbauende Anwendung für die Analyse und Visualisierung von Geodaten, LGPL*  
<http://openev.sourceforge.net/>
- [S4] Boost Community (2013)  
Boost 1.53  
*Freie C++-Bibliothek (insbesondere Template-Bibliothek), Boost Software License*  
<http://www.boost.org>
- [S5] CentOS-Team (2013)  
CentOS 5.9  
*Linux-Distribution, die auf Red Hat aufbaut, u.a. GPL*  
<http://www.centos.org/>
- [S6] David Horman  
Multiblend 0.6  
*Überblenden zwischen Bildern unter Berücksichtigung der Ausdehnung von Bildmerkmalen*  
<http://horman.net/multiblend/>
- [S7] Deutsches Zentrum für Luft und Raumfahrt e.V.  
Earth Observation Center, Institut für Methodik der Fernerkundung  
Xdibias  
*Erweitertes („extended“) Digitales Bildauswertungssystem - Ein proprietäres Bildanalyse- und verarbeitungssystem mit dem Schwerpunkt auf Fernerkundungsdaten.*  
<http://www.dlr.de/eoc/>

- [S8] Dwayne Richard Hipp)  
SQLite 3.3.6 (2006), SQLite 3.7.15.2 (2013)  
*Relationales In-Process Datenbanksystem, gemeinfrei (Public Domain)*  
<http://sqlite.org/>
- [S9] Eric Jones, Travis Oliphant, Pearu Peterson und weitere  
SciPy 0.7.1, SciPy 0.12.0  
*Wissenschaftliches Rechnen mit Python, BSD*  
<http://www.scipy.org/>
- [S10] Éric Lemoine (2013)  
GeoAlchemy 0.7.2  
*Ergänzt SQLAlchemy um Unterstützung für Geodaten, MIT*  
<http://www.geoalchemy.org/>
- [S11] GNU Project  
GCC 4.1.2 (2007), GCC 4.6.3 (2012), GCC 4.8.1 (2013)  
*GNU Compiler Collection - Compiler Suite des GNU-Projekts, GPLv2*  
<http://gcc.gnu.org/>
- [S12] Gentoo Foundation, Inc. (2013)  
Gentoo Linux (Prefix), *Rolling Release*  
*Quellenbasierte Linux-Distribution (Rolling Release Distribution); Prefix-Projekt erlaubt Installation in einem anderen, als dem Wurzelverzeichnis und dadurch insbesondere als Gast in einer bestehenden Linux-Installation, u.a. GPL*  
<http://www.gentoo.org/proj/en/gentoo-alt/prefix/>
- [S13] Itseez (2013)  
OpenCV 2.4.5  
*Programmbibliothek für effiziente Bildverarbeitung und maschinelles Sehen, BSD*  
<http://opencv.org>
- [S14] Michael Bayer (2012)  
SQLAlchemy 0.6.9)  
*Quelloffenes SQL-Toolkit sowie ORM-Mapper für die Python-Umgebung, MIT*  
<http://www.sqlalchemy.org/>
- [S15] Open Source Geospatial Foundation  
GEOS 3.3.0 (2011), GEOS 3.3.8 (2013)  
*C++-Programmbibliothek für die Verarbeitung geometrischer Daten, LGPL*  
<http://geos.osgeo.org/>
- [S16] Python Software Foundation  
Python 2.4.3 (2004), Python 2.7.3 (2010), Python 2.7.5 (2013)  
*Interpreter für die Python-Programmiersprache und Standardlaufzeitbibliothek, Python Software Foundation License*  
<http://www.python.org/>
- [S17] Olga Veksler & Andrew Delong [32], [43]–[45]  
GCO-3.0

*Implementierung der  $\alpha$ -Expansion- und  $\alpha$ - $\beta$ -Swap-Algorithmen (Lösen von Multi-Label Optimierungsproblemen), frei für die wissenschaftliche Nutzung*

<http://www.python.org/>

- [S18] QGIS Development Team, Open Source Geospatial Foundation

QGIS (früher auch als Quantum GIS bezeichnet)

*Quelloffenes Geoinformationssystem (GIS), GPLv2*

<http://www.qgis.org/>

- [S19] Sean Gillies (2011)

Shapely 1.2.8

*Python-Interface für die GEOS-C++-Bibliothek (Verarbeitung geometrischer Daten), BSD*

<https://github.com/Toblerity/Shapely>

- [S20] Travis Oliphant & Community

NumPy 1.2.1, NumPy 1.7.1

*Python Erweiterung, die Unterstützung für große Arrays und wissenschaftliches Rechnen bietet, BSD*

<http://www.numpy.org/>



---

## Literatur

---

- [1] S. Liang, X. Li und J. Wang, *Advanced Remote Sensing: Terrestrial Information Extraction and Applications*, Ser. Academic Press. Academic Press, 2012, isbn: 9780123859549.
- [2] M. Kerschner, „Seamline detection in colour orthoimage mosaicking by use of twin snakes“, *ISPRS journal of photogrammetry and remote sensing*, Bd. 56, Nr. 1, S. 53–64, 2001.
- [3] E. S.-2. Team, „Gmes sentinel-2 mission requirements document“, European space Agency, MRD, 2010. Adresse: [http://esamultimedia.esa.int/docs/GMES/Sentinel-2\\_MRD.pdf](http://esamultimedia.esa.int/docs/GMES/Sentinel-2_MRD.pdf) (besucht am 03.08.2013), final.
- [4] R. Müller, T. Krauß, M. Schneider und P. Reinartz, „Automated georeferencing of optical satellite data with integrated sensor model improvement“, *Photogrammetric Engineering and Remote Sensing*, Bd. 78, S. 61–74, 2012.
- [5] R. Szeliski, „Image alignment and stitching: a tutorial“, *Found. Trends. Comput. Graph. Vis.*, Bd. 2, Nr. 1, S. 1–104, Jan. 2006, issn: 1572-2740.
- [6] —, *Computer Vision: Algorithms and Applications*, 1st. New York, NY, USA: Springer-Verlag New York, Inc., 2010, isbn: 1848829345, 9781848829343.
- [7] P. J. Burt und E. H. Adelson, „A multiresolution spline with application to image mosaics“, *ACM Transactions on Graphics (TOG)*, Bd. 2, Nr. 4, S. 217–236, 1983.
- [8] P. Pérez, M. Gangnet und A. Blake, „Poisson image editing“, *ACM Transactions on Graphics (TOG)*, Bd. 22, Nr. 3, S. 313–318, 2003.
- [9] R. Szeliski, M. Uyttendaele und D. Steedly, „Fast poisson blending using multi-splines“, in *Computational Photography (ICCP), 2011 IEEE International Conference on*, IEEE, 2011, S. 1–8.
- [10] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin und M. Cohen, „Interactive digital photomontage“, *ACM Trans. Graph.*, Bd. 23, Nr. 3, S. 294–302, Aug. 2004, issn: 0730-0301.
- [11] N. Gracias, M. Mahoor, S. Negahdaripour und A. Gleason, „Fast image blending using watersheds and graph cuts“, *Image Vision Comput.*, Bd. 27, Nr. 5, S. 597–607, Apr. 2009, issn: 0262-8856.
- [12] R. Prados, R. Garcia, N. Gracias, J. Escartin und L. Neumann, „A novel blending technique for underwater gigamosaicing“, *Oceanic Engineering, IEEE Journal of*, Bd. 37, Nr. 4, S. 626–644, 2012, issn: 0364-9059.

- [13] E. Fernandez, R. Garfinkel und R. Arbiol, „Mosaicking of aerial photographic maps via seams defined by bottleneck shortest paths“, *Operations Research*, Bd. 46, Nr. 3, S. 293–304, 1998.
- [14] V. Kaibel und M. A. F. Peinhardt, „On the bottleneck shortest path problem“, Konrad-Zuse-Zentrum für Informationstechnik Berlin, ZIB-Report, 2006. Adresse: <http://www.zib.de/Publications/Reports/ZR-06-22.pdf> (besucht am 03.08.2013).
- [15] Wikipedia. (2013). Widest path problem, Adresse: [http://en.wikipedia.org/wiki/Widest\\_path\\_problem](http://en.wikipedia.org/wiki/Widest_path_problem) (besucht am 03.08.2013).
- [16] M. J. Atallah und M. Blanton, Hrsg., *Algorithms and theory of computation handbook: general concepts and techniques*, 2. Aufl. Chapman & Hall/CRC, 2010, isbn: 978-1-58488-822-2.
- [17] Y. Afek und A. Brand, „Mosaicking of orthorectified aerial images“, *Photogrammetric Engineering and remote sensing*, Bd. 64, Nr. 2, S. 115–124, 1998.
- [18] L. Yu, E.-J. Holden, M. C. Dentith und H. Zhang, „Towards the automatic selection of optimal seam line locations when merging optical remote-sensing images“, *International Journal of Remote Sensing*, Bd. 33, Nr. 4, S. 1000–1014, 2012.
- [19] J. Chon, H. Kim und C.-S. Lin, „Seam-line determination for image mosaicking: a technique minimizing the maximum local mismatch and the global cost“, *{ISPRS} Journal of Photogrammetry and Remote Sensing*, Bd. 65, Nr. 1, S. 86–92, 2010, issn: 0924-2716.
- [20] J. Chon und H. Kim, „Determination of the optimal seam-lines in image mosaicking with the dynamic programming (dp) on the converted cost space“, in *Artificial Intelligence and Soft Computing – ICAISC 2006*, Ser. Lecture Notes in Computer Science, L. Rutkowski, R. Tadeusiewicz, L. Zadeh und J. Żurada, Hrsg., Bd. 4029, Springer Berlin Heidelberg, 2006, S. 750–757, isbn: 978-3-540-35748-3.
- [21] F. Bignalet-Cazalet, S. Baillarin und D. Greslou, *Automatic and generic mosaicing of multisensor images: an application to pleiades hr*, 2012.
- [22] P. Soille, „Morphological image compositing“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Bd. 28, Nr. 5, S. 673–683, 2006, issn: 0162-8828.
- [23] P. Soille, „The image-2006 mosaic project“, European Commission - Joint Research Center - Institute for Environment und Sustainability, EUR - Scientific and Technical Research Reports, 2011. Adresse: <http://publications.jrc.ec.europa.eu/repository/handle/111111111/22218>.
- [24] Y. Wan, D. Wang, J. Xiao, X. Lai und J. Xu, „Automatic determination of seamlines for aerial image mosaicking based on vector roads alone“, *{ISPRS} Journal of Photogrammetry and Remote Sensing*, Bd. 76, S. 1–10, 2013, issn: 0924-2716.
- [25] B. Summa, J. Tierny und V. Pascucci, „Panorama weaving: fast and flexible seam processing“, *ACM Trans. Graph.*, Bd. 31, Nr. 4, 83:1–83:11, Juli 2012, issn: 0730-0301.
- [26] C. Han und H. Hoppe, „Optimizing continuity in multiscale imagery“, *ACM Trans. Graph.*, Bd. 29, Nr. 6, 171:1–171:10, Dez. 2010, issn: 0730-0301.

- [27] U. Ansorge und H. Leder, *Wahrnehmung und Aufmerksamkeit*. VS Verlag für Sozialwissenschaften | Springer Fachmedien Wiesbaden GmbH, 2011, ISBN: 9783531920948.
- [28] T. Igarashi, „Alos mission requirement and sensor specifications“, *Advances in Space Research*, Bd. 28, Nr. 1, S. 127–131, 2001.
- [29] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen und C. Rother, „A comparative study of energy minimization methods for markov random fields with smoothness-based priors“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Bd. 30, Nr. 6, S. 1068–1080, 2008.
- [30] O. Veksler, „Efficient graph-based energy minimization methods in computer vision“, Diss., Cornell University, 1999.
- [31] Y. Boykov und O. Veksler, „Graph cuts in vision and graphics: theories and applications“, in *Handbook of mathematical models in computer vision*, Springer, 2006, S. 79–96.
- [32] Y. Boykov und V. Kolmogorov, „An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Bd. 26, Nr. 9, S. 1124–1137, 2004.
- [33] D. Jungnickel, *Graphs, Networks and Algorithms*, 4th. Springer Publishing Company, Incorporated, 2012, ISBN: 3642322778, 9783642322778.
- [34] S. Palsule, B. Shah, G. Paswan, A. Sharma, A. Garg, H. Pandya, S. S. Srivastava und S. Roy, „Resourcesat-1 data quality evaluation system“, *International Journal of Applied Earth Observation and Geoinformation*, Bd. 10, Nr. 2, S. 147–158, 2008.
- [35] L. Vincent und P. Soille, „Watersheds in digital spaces: an efficient algorithm based on immersion simulations“, *IEEE transactions on pattern analysis and machine intelligence*, Bd. 13, Nr. 6, S. 583–598, 1991.
- [36] *Beschluss der Kommission vom 12. Dezember 2011 über die Weiterverwendung von Kommissionsdokumenten*, 2011/833/eu, Europäische Kommission, 2011. Adresse: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:330:0039:0042:de:PDF>.
- [37] R. C. Gonzalez und R. E. Woods, *Digital Image Processing (3rd International Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2008, ISBN: 013505267X.
- [38] S. Di Zenzo, „A note on the gradient of a multi-image“, *Computer Vision, Graphics, and Image Processing*, Bd. 33, Nr. 1, S. 116–125, 1986.
- [39] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel und G. Randall, „Lsd: a fast line segment detector with a false detection control“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Bd. 32, Nr. 4, S. 722–732, 2010.
- [40] E. P. Baltsavias, S. Kocaman, D. Akca, K. Wolff, E. P. Baltsavias, E. P. Baltsavias, D. Akca und D. Akca, *Geometric and radiometric investigations of Cartosat-1 data*. Citeseer, 2007.

- [41] R. Müller, T. Krauß, M. Lehner, P. Reinartz, J. Forsgren, G. Rönnbäck und Å. Karlsson, „Image2006 european coverage – methodology and results“, *GMES fast track land service*, 2009. Adresse: [http://earth.esa.int/pub/ESA\\_DOC/Image2006-v1\\_01.pdf](http://earth.esa.int/pub/ESA_DOC/Image2006-v1_01.pdf).
- [42] D. Brunner und P. Soille, „Iterative area seeded region growing for multichannel image simplification“, in *Mathematical Morphology: 40 Years On*, Springer, 2005, S. 397–406.
- [43] Y. Boykov, O. Veksler und R. Zabih, „Fast approximate energy minimization via graph cuts“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Bd. 23, Nr. 11, S. 1222–1239, 2001.
- [44] V. Kolmogorov und R. Zabih, „What energy functions can be minimized via graph cuts?“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Bd. 26, Nr. 2, S. 147–159, 2004.
- [45] A. DeLong, A. Osokin, H. N. Isack und Y. Boykov, „Fast approximate energy minimization with label costs“, *International Journal of Computer Vision*, Bd. 96, Nr. 1, S. 1–27, 2012.



---

# **Eigenständige Anfertigung der Arbeit**

---

## **Erklärung über die eigenständige Anfertigung der Arbeit**

Ich erkläre hiermit, dass die vorliegende Masterarbeit mit dem Titel

### **Vollautomatisches Mosaikieren von Satellitenbildern**

eigenständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet wurden.

---

Philipp Serr  
München, den 30.09.2013